Defending Against Phishing Attacks

Xun Dong

Submitted For The Degree of Doctor of Philosophy

The University of York

Department of Computer Science

September 2009

For My Mum and Dad

Abstract

Valuable information, such as user authentication credentials and personal sensitive information, can be obtained by exploiting vulnerabilities within the user's understanding of a system, and particularly a lack of understanding of the user interface.

As the barrier to exploiting system vulnerabilities has increased significantly with time, attacking users has rapidly become a more efficient and effective alternative.

To protect users from phishing attacks system designers and security professionals need to understand how users interact with those attacks. In this thesis I present an improved understanding of the interaction and three novel mechanisms to defend against phishing attacks.

Contents

List of Tables

List of Figures

Acknowledgements

The author would like to thank Professor John Andrew Clark and Dr Jeremy Jacob for their kind support and encouragement to complete this thesis including help tidying the English. I would like to thank my wife Ji Xiaoyan, my daughter Dong Yue, and my parents; without their support this thesis would not be possible. I also should like to thank Dr Chen Hao and Tom Haines for discussion about my ideas and also for their kindly suggestions. I wish to thank all the people who have helped me, and especially the Engineering and Physical Sciences Research Council (EPSRC) of the United Kingdom for their sponsorship of the project "Defending the Weakest Link: Intrusion via Social Engineering"(EPSRC Grant EP/D051819/1). I am grateful also to the University of York's Department of Computer Science.

Xun Dong

Author's Declaration

This thesis is the work of Xun Dong and was carried out at the University of York, United Kingdom. Work appearing here has appeared in print as follows:

- Modelling User-Phishing Interaction. Xun Dong, John A Clark and Jeremy L Jacob. Human System Interaction, 2008.

- Threat Modelling in User Performed Authentication. Xun Dong, John A Clark and Jeremy Jacob. 10th International Conference on Information and Computer Security, 2008.

- Detection of Phishing Websites by User Behaviours. Xun Dong, Jeremy Jacob and John A Clark. International Multi-conference on Computer Science and Information Technology, 2008.

- Defending the Weakest Link: Detection of Phishing Websites by User Behaviours. Xun Dong, Jeremy Jacob and John A Clark. Telecommunication Systems 45(2-3): 215-226 (2010).

The collection of social engineering attacks examples can also be found on the web site: http://www-users.cs.york.ac.uk/~xundong/se/se_attacks.php

Chapter 1

Introduction

This chapter describes what phishing attacks are and why it is so important that we must defend against them effectively. It also explains why by improving our understanding of the users' psychological models and fundamentals of phishing attacks, more effective countermeasures can be inspired.

1.1 What is a Phishing Attack?

While the Internet has brought unprecedented convenience to many people for managing their finances and investments, it also provides opportunities for conducting fraud on a massive scale with little cost to the fraudsters. Fraudsters can manipulate users instead of hardware/software systems, where barriers to technological compromise have increased significantly. Phishing is one of the most widely practised Internet frauds. It focuses on the theft of sensitive personal information such as passwords

and credit card details. Phishing attacks take two forms:

- attempts to deceive victims to cause them to reveal their secrets by pretending to be trustworthy entities with a real need for such information;

- attempts to obtain secrets by planting malware onto victims' machines.

The specific malware used in phishing attacks is subject of research by the virus and malware community and is not addressed in this thesis. Phishing attacks that proceed by deceiving users are the research focus of this thesis and the term 'phishing attack' will be used to refer to this type of attack.

Despite numerous countermeasure efforts, the scale and sophistication of phishing attacks are still increasing. The number of reported phishing web sites increased 50 percent from January 2008 to January 2010 [73]. During the 2008 world financial crisis phishing attack incidents increased three times compared to the same period in 2007. The real figure could be much higher because many sophisticated phishing attacks (such as context aware phishing attacks, malware based phishing attacks, and real-time man-in-the-middle phishing attacks against one-time passwords [79]) may not all have been captured and reported. Victims of these phishing attacks may never realise they have been attacked, and many of these sophisticated attacks are targeted and small scale, hence it is likely many of them will not have been captured and reported.

Phishing attacks have not only caused significant financial damage to both users and companies/financial organizations, but also have damaged users' confidence in e–commerce as a whole. According to Gartner analysts, financial losses stemming from phishing attacks rose to more than 3.2 billion USD with 3.6 million victims in 2007 in the US [60], and consumer anxiety about Internet security resulted in a two billion USD loss in e–commerce and banking transactions in 2006 [58]. In the United Kingdom losses from web banking frauds (mostly from phishing) almost doubled to $46m in 2005, from $24m in 2004, while 1 in 20 computer users claimed to have lost out to phishing in 2005 [60]. [1]

As the Internet continues to transform how people manage their data, complete their business tasks, and share resources, the value of user authentication credentials to access those services will increase. Phishing attacks may compromise the integrity of such valuable authentication credentials, and must be defended against effectively and efficiently.

From the victim's point of view, a phishing attack can be broken down into three stages:

1. Attacker approach: the approach by attackers on a chosen communication channel;

2. Interaction: interaction with the fraudulent entity which impersonates its legitimate counterpart;

3. Exploitation: exploitation of the obtained secret information for financial gain.

[1] These figures are the latest version author can obtain on 1st June 2010.

A typical phishing attack would engage victims via emails, then lead them to a phishing website. Attackers can either directly use the obtained user authentication credentials to raid victims' financial assets, or sell them to other criminals. Here I describe a real-life phishing attack to illustrate how phishing works.

The Anti-Phishing Working Group (APWG) [73] and Phishtank [74] collect and archive a large number of reported phishing attacks. An example from Phishtank is an attack against HBOS bank customers on 15th January 2009. It happened during the banking crisis when the HBOS banking group was about to be taken over by Lloyds TSB banking group.

In stage one: the potential victims received an email (shown in Figure 1.1), which claimed to be from HBOS, asking customers to check how the acquisition would affect their bank accounts and update personal information if necessary through the provided hypertext link.

In stage two: if users believed they were interacting with a legitimate email and followed the provided hypertext link, they would give away their authentication credentials to the phishing website (shown in Figure 1.2).

In stage three: the attackers would sell the authentication credentials to others or directly use them to transfer money away from victims' accounts.

HBOS customers could very easily be deceived by this phishing email. At the time the acquisition was widely reported by public media and the deal was set to be concluded on 19th January 2009. As a result HBOS customers might well have expected such communications. RBS is one of

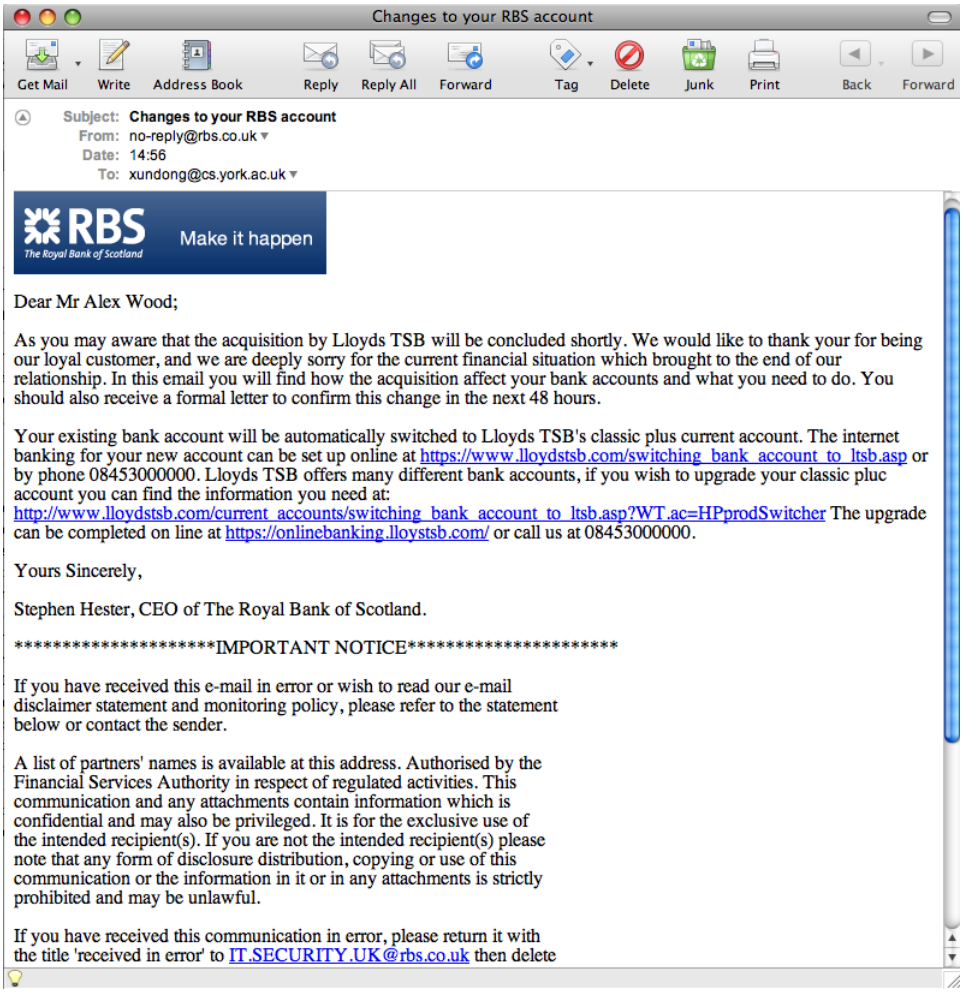


Figure 1.1: Phishing Email Screen Shot

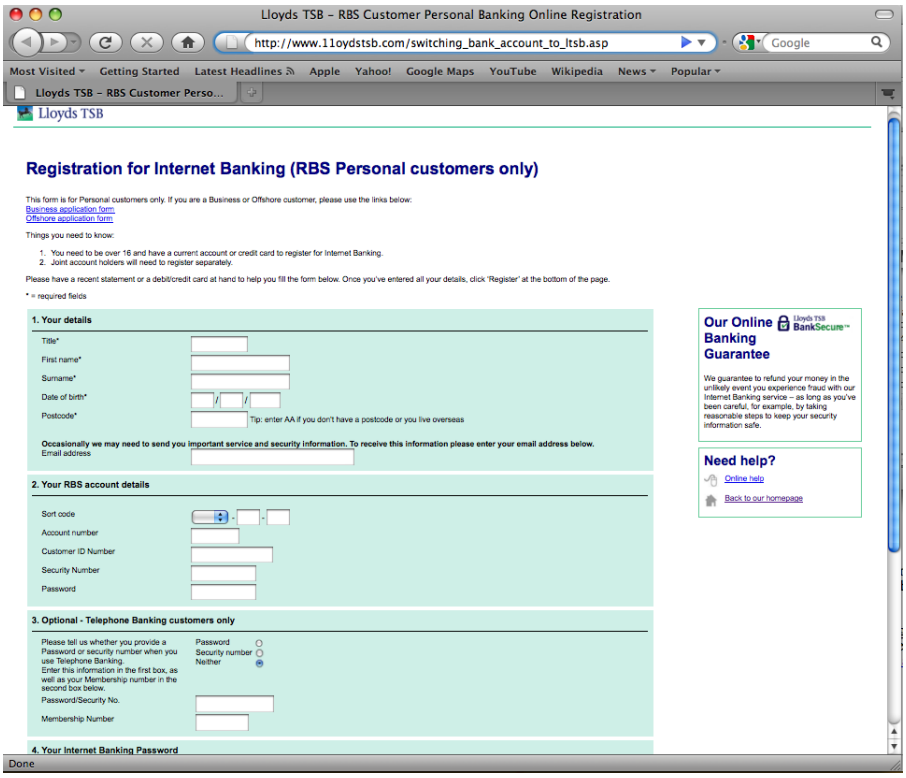


Figure 1.2: Phishing Website Screen Shot

the banks owned by the HBOS group. In addition, the action suggested in this email might seem both rational and professional. The email header also suggests it is from RBS – the 'From' field is no-reply@rbs.co.uk. The hypertext links in the emails except the one leading to the phishing website all link to the legitimate Lloyds TSB website. The phishing website has been carefully prepared to have the same style and layout as the legitimate Lloyds TSB website, and all the hypertext links are linked to the legitimate website. Only users who carefully examine the domain name of the website would discover they are visiting a phishing website; the digits '11' (one one) look very similar to the letters 'll' at a glance.

Users do not know when they will be attacked. To avoid falling victim to this attack, users must either analyse the IP address from which an email is actually sent from or consistently check very carefully the URL strings of the hypertext links.

1.2 The Thesis

1.2.1 Statement and the Interpretation of the Hypothesis

The thesis is:

A more refined understanding of the nature of deception in phishing attacks would facilitate more effective user-centred threat identification of web based authentication systems, the development of countermeasures to identified threats, and the production of guidelines for phishing–resilient system designs.

This thesis is an example of multi-disciplinary research where human computer interaction and security meet. Essentially a phishing attack aims to engineer a false perception within a victim's mind. Having had a false perception constructed in his mind the victim will carry out actions to satisfy the attacker's goals. To defend against such attacks effectively and efficiently, an understanding of how human users interact with phishing attacks at the user interface and how users perceive the information presented at the interface to form the mental model is vital. Some user studies [21, 77, 26, 50, 46, 88] have investigated human factors, but greater insights are still needed.

An improved understanding of human computer interaction in this domain can aid both prevention and detection of phishing attacks. In the prevention area, the knowledge can help system designers choose user interfaces that help users form accurate mental models, and hence make appropriate decisions; the knowledge can also be used to analyse an authentication system for the vulnerabilities that attackers might exploit to carry out phishing attacks (existing threat modelling techniques do not address the phishing threat, because they do not consider usability of the system and their human factors). This knowledge could also be applied to design detection systems that are easier for users to understand while being much harder for attackers to bypass.

This thesis will demonstrate that all above claims can be achieved. There are three target audiences for the research reported here: the anti-phishing research community, system designers who design and implement user interfaces for authentication systems, and security practitioners who analyse existing systems and provide security education for end users.

1.2.2 Research Method

Overall the work described in this thesis follows a constructive research approach. The research started with studying literature bodies of: social engineering, phishing attack techniques, human factors in phishing attacks, and phishing attacks countermeasures. The author concluded that more research to understand how users make decisions during phishing attack interactions was needed. Users' decision making plays a vital role in deciding the outcome of a phishing attack. With better understanding in this area, more effective countermeasures could be discovered.

In the next phase of the research cognitive walkthroughs [13] were used to study a large number of social engineering and phishing attack incidents. Drawing on the findings of the walkthoughs, a user-phishing interaction model was constructed.

The knowledge obtained in the first two phases of the research formed the base for the final phase of the research – creating more effective phishing attacks prevention and detection methods. The author proposed a threat modelling method to identify threats that can be realised by attacking users of authentication systems. To demonstrate the merits of the method, it is applied to study two widely used authentication systems. The user-phishing interaction model suggests users, who fall victims to phishing attacks, construct false perceptions in their minds and subsequently carry out actions to release sensitive information to attackers. The false perceptions and subsequent actions are common to most, if not all, phishing attacks. This inspired the creation of a detection technique which ignores how phishing attacks are presented, but rather focuses on users' actions to release sensitive information to parties to

whom such sensitive information has never been released before. The findings in the first two phases of the research have also influenced the design decisions relating to the usability of this detection system. The detection accuracy (including false positives) of the detection system is also evaluated.

1.3 Major Contributions

The major contributions this thesis makes are:

User Phishing Interaction Model: a psychological model to capture the general process of decision making during user-phishing interaction and important factors that can influence the outcome of such decision making. It is useful for designing security tools/indicators, evaluating how well a phishing detection tool can assist users to detect phishing attacks, and designing effective and efficient user education methods.

Threat Modelling methods for Web Authentication Systems: a framework and related methods to identify and assess user-related vulnerabilities within internet based user authentication systems.

User Behaviour Based Phishing Attacks Detection System: a novel phishing website detection approach and its prototype.

1.4 Brief Overview of the Chapters

The subsequent chapters of this thesis are as follows:

Chapter 2 reviews existing publications in the fields of understanding human factors in phishing attacks, phishing detection systems, and social engineering/semantic attacks in general. It also identifies gaps in existing work.

Chapter 3 describes a model to capture essential characteristics within user–phishing–attack interactions, and describes the applications of this model.

Chapter 4 introduces a new method to systematically analyse potential vulnerabilities that exist in web-based authentication systems. It also presents two case studies to demonstrate the merit of this method.

Chapter 5 describes the design, implementation and evaluation of UBPD – a phishing website detection system. The detection system is based on past user behaviours and it is much harder to bypass than most current detection systems.

Chapter 6 concludes the thesis and its contributions and also points out areas where future research could be conducted.

Chapter 2

Introduction to Social Engineering and Phishing attacks

This chapter provides an introduction to social engineering and phishing attack techniques, and reviews related human factors studies and techniques to counter phishing attacks.

2.1 Overview

The research literature reviewed in this chapter can be classified into the following four categories:

1. understanding of attacks (both social engineering attacks in general and phishing attacks in particular);

2. bounded rationality decision making theory;

3. investigation of human factors in security; and

4. techniques to prevent and detect phishing attacks.

2.2 Understanding of Attacks

2.2.1 Social Engineering/Semantic Attacks

Social engineering (SE) attacks generally achieve their goals by manipulating victims to execute actions against their interests. This term typically applies to trickery or deception for the purpose of information gathering, fraud or gaining computing system access. Phishing attacks are a subset of social engineering attacks.

Kevin Mitnick, who acquired millions of dollars by carrying out social engineering attacks, is arguably the best known social engineering attacker. His book "The art of deception: Controlling the Human Element of Security" [65] defined social engineering as follows:

> Using influence and persuasion to deceive people by convincing them that the attacker is someone he is not, or by manipulation. As a result, the social engineer is able to take advantage of people to obtain information, or to persuade them to perform an action item, with or without the use of technology.

There is no commonly accepted definition for the term " Social Engin-

eering". Mitnick's definition has considerable appeal [1]. It highlights that people are the major target of attack, indicates some of the important tools used by attackers, such as influence and persuasion, and summarises the objectives of the attack. But this definition does not address *why* people can be so easily deceived, and does not provide a fundamental structure or model for SE attacks. Moreover, the objectives of SE in his description are not comprehensive.

Mitnick's book has four parts. Part 1 introduces the basic elements of social engineering. Parts 2 and 3 use a lot of "fictional" stories and phone transcripts to show how an attacker can manipulate employees into revealing seemingly innocent pieces of information that are later used (sometimes on an ongoing basis) to extend the confidence trick, gain more access, steal information, "borrow" company resources, and otherwise defraud companies or individuals out of just about anything. The stories are very basic examples of social engineering that are designed to raise awareness. The majority of the tactics described focus on impersonating someone who should have legitimate access to the data, but for one reason or another cannot get to it. The attacker then enlists the aid of a helpful but unsuspecting employee to retrieve the information for them. In many cases, this is a process that involves a number of employees, all of whom provide small bits of seemingly unimportant information that become pieces in a large puzzle. He also analyses the attacks from

[1] Other definitions: "The art and science of getting people to comply to your wishes" – Harl "People hacking"[39]; "social engineering is the process of deceiving people into giving confidential, private or privileged information or access to a hacker." –Rusch, Jonathan J. [76]; "social engineering is generally a hacker's clever manipulation of the natural human tendency to trust. The hacker's goal is to obtain information that will allow him/her to gain unauthorized access to a valued system and the information that resides on that system." – Sarah Granger[34, 35]

both the attacker's and victim's perspective and offers advice on how to protect against similar attacks. In Part 4 Mitnick provides a number of sample security policies and procedures, including data classification categories, verification and authentication procedures, guidelines for awareness training, methods of identifying social engineering attacks, warning signs, and flowcharts for responding to requests for information or action. The majority of policies and procedures are not novel and are largely based on the ideas suggested by Charles Cresson Wood [15].

Many publications [33, 34, 35, 36, 59] on SE have summarised the techniques SE uses and the media through which SE is conducted. A few of them have tried to classify SE attacks. One of the best known classifications of SE is given by Sarah Granger [34, 35]. It partitions social engineering into the following five categories:

1. Social engineering by phone. (Telephone communication);

2. Dumpster diving. (Office waste);

3. Online social engineering. (The Internet);

4. Persuasion. (Face to face communication); and

5. Reverse social engineering

It classifies SE based on the techniques rather than the nature of the attacks. The first four categories are based on the communication medium used to convey SE. (The communication medium is given in parentheses above). The last category is a special case of SE using scenarios where

the victim is the party who initiates the communication. In such a case the victim will be much easier to deceive, because they initiated the communication and they will likely trust the attacker more. Others [33, 59] have chosen to classify SE by targets of attacks, or by tools or techniques used by the attacks.

All these classifications are useful for introducing SE. However, they do not reveal the nature of SE nor provide any insights of why SE works. We might expect those existing classifications to fail to cover new attacks as SE evolves, especially when new SE attacks use different communication media, or are different in appearance. For example, the USB SE attacks [20] do not fit into Granger's classification. Most importantly such classifications cannot directly be applied to facilitate proactive SE detection and guide the design of SE resilient systems. Existing classifications view social engineering from the attacker's point of view. They are useful to define what SE is, and serve as a tutorial about how SE attacks are executed. But they are less useful when it comes to help identify SE attacks, improve the security of the system at the design stage, and contribute to automated detection solutions.

2.2.2 Phishing Attacks

Phishing is a special type of social engineering attack. In his phishing attacks guides [70, 69] Ollmann has described the anatomy of phishing attacks and surveyed phishing attack prevention techniques. He described phishing attack threats from the following three aspects:

- social engineering factors;

- how phishing messages are delivered to victims via email, web, IRC, instant messenger, and trojan horses;

- techniques used in phishing attacks such as man-in-the-middle attacks, URL Obfuscation, cross site scripting, preset session attacks, etc.

In his report he also provides detailed advice on how to use existing technologies to counter phishing threats from both client and server sides as well as on what organisations can do to prevent them. He identifies the following countermeasures that can be applied on the client side:

1. desktop protection technologies;

2. utilisation of appropriate, less sophisticated communication settings;

3. user application-level monitoring solutions;

4. locking-down browser capabilities;

5. digital signing and validation of email; and

6. improving general security awareness.

He also identifies the following countermeasures that can be applied on the server side:

1. improving customer awareness;

2. providing validation information for official communications;

3. ensuring that Internet web applications are securely developed and doesn't include easily exploitable attack vectors;

4. using strong token-based authentication systems; and

5. keeping naming systems simple and understandable.

Finally he also suggests businesses and ISP's should use technologies to protect against phishing attacks at the enterprise-level. The following enterprise solutions are suggested:

1. automatic validation of sending email server addresses;

2. digital signing of email services;

3. monitoring of corporate domains and notification of "similar" registrations;

4. perimeter or gateway protection agents; and

5. third-party managed services.

Together with the counter-measure mechanisms on both client and server sides, phishing attacks can be defended effectively at multiple levels, giving better protection to users.

Watson et al. have carried out a study to observe real phishing attacks in the wild by using Honeynet [84]. This study focuses on how attackers build, use and maintain their infrastructure of hacked systems. The report

is based on data collected by the German Honeynet Project and the UK Honeynet Project. They do not cover all possible phishing methods or techniques, focussing instead on describing the follow three techniques observed:

1. phishing through compromised web servers;

2. phishing through port redirection; and

3. phishing using botnets.

They also briefly describe how the observed attacks transfer money they have stolen from victims' bank accounts. Their work provides some insights into how phishing attacks are implemented in reality.

To formally understand the phishing attack from the technical point of view, Jacobsson has introduced a method to describe a variety of phishing attacks in a uniform and compact manner via a graphical model [47]. He has also presented an overview of potential system vulnerabilities and corresponding defence mechanisms.

> In such a graph, there are two types of vertices; those corresponding to access to some information; and those corresponding to access to some resource. Actions are represented as edges in the graph. Two vertices are connected by an edge if there is an action that would allow an adversary with access corresponding to one of the vertices to establish access corresponding to the other of the vertices. Some set of nodes correspond to possible starting states of attackers, where the state contains all information available to the attacker. (This

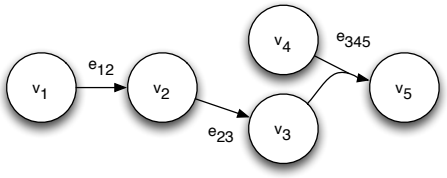


Figure 2.1: Graph Model For A Man-in-a-middle Phishing Attack From [47]

> may simply consist of publicly available information.) One node corresponds to access to some resource of the attacker's choosing, call this the target node. For an attack to be successful, there needs to be a path from a starting state to the target node. Figure 2.1 illustrates the graphical model for a man-in-a-middle phishing attack.

An explanation [47] for the model illustrated in Figure 2.1 is given below:

> vertex v1 corresponds to knowledge of the name of the administrative contact of a domain to be attacked. Vertex v2 corresponds to knowledge of the appropriate credit card number, and vertex v3 to access to the account. Finally, v4 corresponds to knowledge of a service for which a user in the attacked domain is registered as the administrative contact, and where passwords are emailed to administrators claiming to have forgotten the passwords, and v5 to access to the account of such a site. There is an edge e12 corresponding to the action of finding out credit card numbers associated with a person with a given name. Edge e23 corresponds to the action

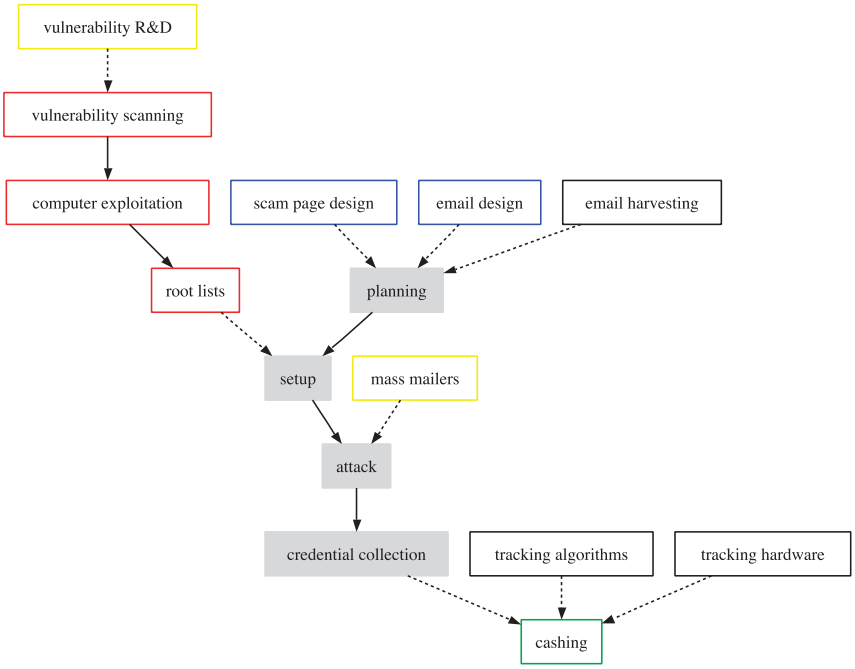


Figure 2.2: Phishing Attacks From Start To Finish [6]

of using the correct credit card number to authenticate to the site, and edge e345 to requesting a forgotten password to be emailed. Note that both v3 and v5 may be considered target nodes

Although the model can be consistently applied to describe phishing attacks, it offers little value in helping understand why phishing attacks work and how to prevent and detect them.

Abad has studied the flow of phishing attacks from an economics point of view[6]. He derived the reported results by analysing 3,900,000 phishing e-

mails and 220,000 messages. The data was collected from 13 key phishing-related chat rooms and 48,000 users which were spread across six chat networks and 4,400 compromised hosts used in botnets. He concludes that phishing attacks from the attackers' point of view have five stages: planning, setup, attack, collection, and cashing. (The graphical model is illustrated in Figure 2.2.) He also discovered that phishing attacks are organized and well co-ordinated with participants having specific roles to play. These participants serve each other by exchanging services or information for cash and their behaviours follow the laws of supply and demand. Using the model presented by this study, one can clearly understand the sophistication of phishing attacks. The model could also be useful for identifying points where intervention could be made to prevent phishing attacks from succeeding.

In-Session Phishing Attacks

The phishing attacks which have been described so far all need to actively engage users via a communication channel. In–session phishing [54], a more recently reported type of attack, uses a more passive mode, and yet is still very effective.

This type of attack exploits user's opening of multiple web pages at the same time. It can succeed if the users have logged into one of the websites which the attacker would like to impersonate and have opened a web page from a compromised website. On the compromised website the attacker plants malware to identify which website the victim user is currently logged on to, then the malware presents a dialogue box, which asks the user to retype their user name and password because the session

has expired, or complete a customer satisfaction survey, or participate in a promotion, etc. Since the user had recently logged onto the targeted website, he/she is unlikely to suspect this pop-up is fraudulent and thus is likely to provide the requested details.

Identifying websites to which a user is currently logged onto can be more difficult to achieve. Jeremiah Grossman et al. have described a method to detect the stage of authentication by loading images that are only accessible to logged-in users [19]. There are other methods that can achieve this by exploiting vulnerabilities within web browsers. However, those methods are not general. In Section 2.4.1, a general method is described.

2.3 Bounded Rationality

Phishing attacks achieve their goals when users have been deceived to carry out certain actions. It is certainly against users' interests to satisfy attackers' goals. However, they still decide to do so. If human behaviour can be understood as a purposeful attempt to achieve well-being, then why would phishing attack victims make such decisions?

Bounded rationality [80] is the decision making theory proposed by Herbert Alexander Simon. Simon suggested that decision-makers arrive at their decisions by rationally applying the information and resources that are easily available to them, with the consequence that satisfactory rather than optimal decisions result.

Bounded rationality theory has great value for understanding why users make certain decisions during their interactions with phishing attacks. It recognises that in practice rational decisions are often impossible and users' rationality is limited by information available to them. In phsihing attacks, rationality of users could be strongly limited by the information presented to them at the user interface. It also recognises that the time available to decision makers and their own cognitive ability are limiting factors. In Simon's theory, the cost of gathering and processing the information would also greatly influence the rationality of a decision one made. It would be interesting to apply the principles of bounded rationality to understand user victims' decision making during interactions with phishing attacks.

2.4 Human Factors

In phishing attacks human users are the targets of attack. To be able to provide them with appropriate warning messages and design secure usable interfaces, understanding why they fall victim and how they behave in cyberspace is essential.

Dhamija et al. have investigated why users fall victim to phishing attacks by carrying out a controlled phishing attack user study [21]. In this study 20 web sites were presented in no particular order to 22 participants. The participants were asked to determine which websites they visited were fraudulent, and to provide rationales. They identified three major causes for victimhood:

1. a lack of understanding of how computer systems work. Many users lack underlying knowledge of how operating systems, applications, email and the web work and how to distinguish among these;

2. a lack of attention to security indicators or the absence of security indicators; and

3. the high quality visual deception practised by the phishers.

The highly controlled nature of this study may lead to biased conclusions or failure to identify important factors in why phishing works. In the experiment, users' attention is directed to making a decision regarding the authenticity of the web-sites. However, in a real-world setting, users would have a range of tasks they wish to perform and establishing authenticity of any accessed websites might not be a primary concern.

Schechter et al. evaluated website authentication measures that are designed to protect users from phishing attacks [77]. 67 bank customers were asked to conduct common online banking tasks. Each time they logged in, they were presented with increasingly alarming clues that their connection was insecure. First, HTTPS indicators were removed; second, the participant's site-authentication image (the customer-selected image that many websites now expect their users to verify before entering their passwords) were removed; finally, the bank's password-entry page was replaced with a warning page. After each clue, researchers then checked whether participants entered their passwords or withheld them. The researchers also investigated how a study's design affects participant behaviour: they asked some participants to play specially created user roles and others to use their own accounts and passwords. Their major findings are:

1. users will enter their passwords even when HTTPS indicators are absent;

2. users will enter their passwords even if site authentication images are absent;

3. site-authentication images may cause users to disregard other important security indicators; and

4. role-playing participants behaved significantly less securely than those using their own passwords.

Again because of the experiment conditions, there could be an overestimate of the ineffectiveness of the security indicators.

Egelman et al. examine the effectiveness of web browsers' phishing warnings and examine if, how, and why they fail users [26]. In their study they used a spear phishing attack to expose users to browser warnings. 97% of sixty participants fell for at least one of the phishing messages sent to them; 79% of participants paid attention to an active warning, in contrast only one participant noticed a passive warning. Egelman et al. also applied the C-HIP model [86] (Figure 2.3) from the warning sciences to analyse how users perceive warning messages and suggest:

1. interrupting the primary task: phishing indicators need to be designed to interrupt the user's task;

2. providing clear choices: phishing indicators need to provide the user with clear options on how to proceed, rather than simply

displaying a block of text;

3. failing safely: phishing indicators must be designed such that one can only proceed to the phishing website after reading the warning message;

4. preventing habituation: phishing indicators need to be distinguishable from less serious warnings and used only when there is a clear danger; and

5. altering the phishing website: phishing indicators need to distort the look and feel of the website such that the user does not place trust in it.

The suggestions made by Egelman et al. are very useful indeed, however, their claim on spear phishing could be made more convincing if their study included an extended range of speared phishing attacks. Otherwise, one could also argue that the results exhibit biases due to the small number of attack incidents used or the sophistication of the attacks used in the study.

Jakobsson et al. have studied what makes phishing emails and web pages appear authentic [50]. Elsewhere Jakobsson summarised comprehensively what typical computer users are able to detect when they are carefully watching for signs of phishing [48]. The findings are are:

1. spelling and design matter;

2. third party endorsements depend on brand recognition;

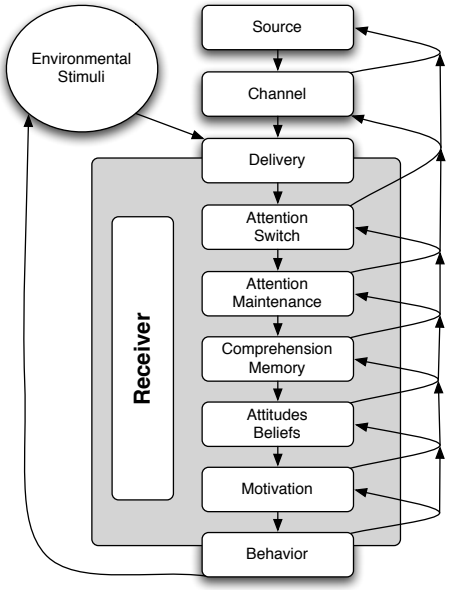


Figure 2.3: C-HIP Model [86]

3. too much emphasis on security can backfire;

4. people look at URLs;

5. people judge relevance before authenticity;

6. emails are very phishy, web pages are a bit phishy, and phone calls are not;

7. padlock icons have limited direct effects; and

8. independent communication channels create trust.

These outcomes provide some comfort and yet are a source of considerable worry, highlighting various opportunities and means of attack. That people look at URLs is a good thing. However, the reason why users look at URLs is not stated, and the degree of attention they pay to them is unclear. The padlock would generally be viewed by many as a significant security mechanism. Not by users, it would appear. The outcome related to media/channel highlights the fact that phishers make highly effective channel choices.

Jagatic et al. have shown how publicly available personal information from social networks (such as Friendster, Myspace, Facebook, Orkut, and Linkedin) can be used to launch effective context aware phishing attacks [46]. In their studies they first determine a victim's social networks and then masquerade as one of their social contacts to create an email to the victim (using email header spoofing techniques). Figure 2.4 illustrates the details of the set up of the study. Their study has shown that not only is

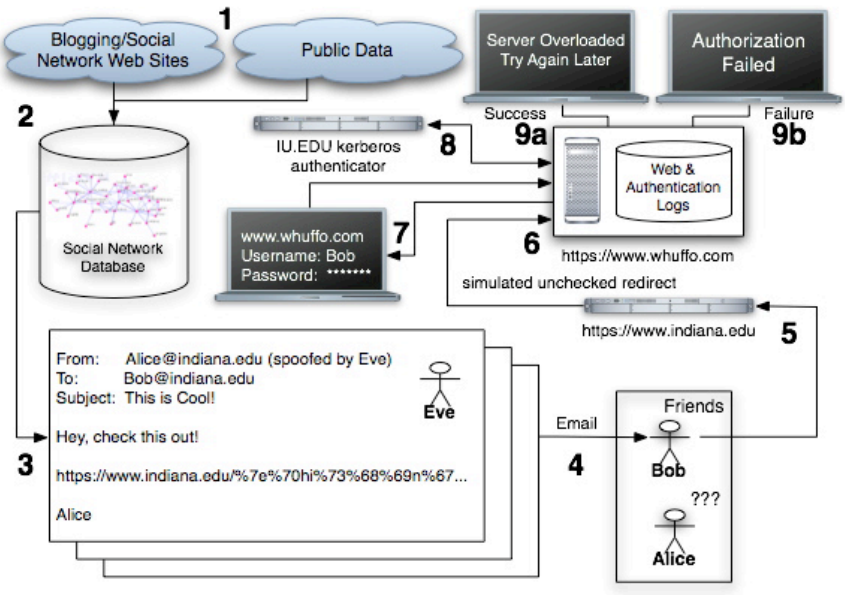


Figure 2.4: Context Aware Phishing Attacks Experiment Design [46]

it very easy to exploit the social network data available on the Internet, but it also increases the effectiveness of the attack significantly. In their experiment, the attacks that took advantage of social networks were four times as likely to succeed.

Below is an explanation of Figure 2.4 (directly taken from [46]):

> 1. Blogging, social network, and other public data is harvested; 2. data is correlated and stored in a relational database; 3. heuristics are used to craft "spoofed" email message by Eve "as Alice" to Bob (a friend); 4. message is sent to Bob; 5. Bob follows the link contained within the email and is sent to an unchecked redirect; 6. Bob is sent to attacker whuffo.com site;

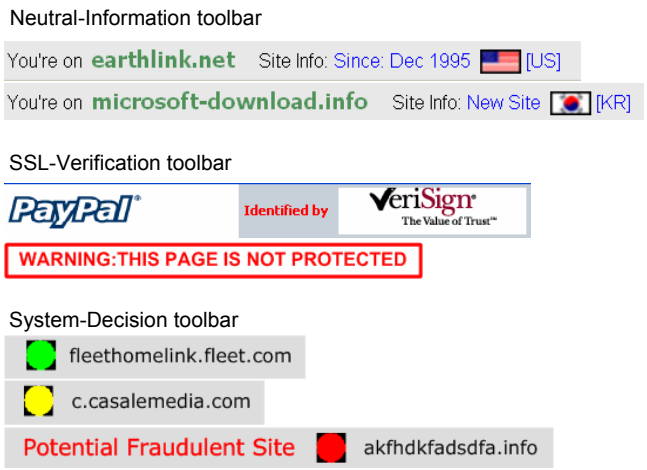


Figure 2.5: Three Simulated Toolbars [88]

7. Bob is prompted for his University credentials; 8. Bob's credentials are verified with the University authenticator; 9a. Bob is successfully phished; 9b. Bob is not phished in this session; he could try again.

Wu et al. have discovered by conducting two user studies that the security tools such as security toolbars are not effective enough to protect people from falling victim to phishing attacks [88]. Features of five toolbars are grouped into three simulated toolbars. The three simulated toolbars shown in Figure 2.5 are: the Neutral Information toolbar, the SSL-Verification toolbar, and the System-Decision toolbar.

In the user study researchers set up dummy accounts in the name of "John Smith" at various legitimate e-commerce websites and then asked the participants to protect those passwords. The participants played the

role of John Smith's personal assistant and were given a printout of John's profile, including his fictitious personal and financial information and a list of his user names and passwords. The task was to process 20 email messages, most of which were requests by John to handle a forwarded message from an e-commerce site. Each message contained a link for the user to click. Some messages are carefully prepared phishing attacks. The researchers then study the participants' response when using various toolbars. Most participants fall victim to the phishing attacks. Based on their findings, the authors suggest that:

1. the alert should always appear at the right time with the right warning message;

2. user intentions should be respected, and if users must make security-critical decisions they should be made consciously; and

3. and it is best to integrate security concerns into the critical path of their tasks so that users must address them.

The user study set up by Wu et al. may lead the users to behave less securely, because the account used is artificial and there are no negative consequences for the participants. Under those conditions users may behave differently than they normally do with their own accounts.

Florencio et al. have carried out a large scale study of password use and password reuse habits [28]. Their study involves half million users over a three month period. Software on the client machine recorded the password usage, strength, and use frequency, etc. They estimated the average number of distinct passwords of a user was 7, and on average

each password is used in 6 different websites. Weak passwords are reused more often than strong passwords. A user on average has over 25 password accounts. Users do use stronger passwords for more important accounts. They also discovered that even as users perceive the need, or are forced, to use stronger passwords, it appears that they use longer lower-case passwords and use upper case and special characters hardly at all. Users appear to forget passwords and perform other administrative functions (reset or change password) a lot. For example, Yahoo password change operations occur 15% as frequently as Yahoo sign-in operations.

Downs et al. conducted interviews with 20 non-expert computer users to reveal their strategies and understand their decisions when encountering possibly suspicious emails [24]. They have found:

- average users are more aware of the threats to computers and confidential information caused by malware than by social engineering attacks;

- Users do pay attention to security indicators but lack sufficient knowledge to correctly interpret them. Interpretation of URL strings is a good example.

- Most user strategies to decide the trustworthiness of email are based on the content of the email. Again this shows users' awareness of threats but lack of knowledge to make correct judgements given current user interface design.

For many years malware such as viruses received a great deal of attention. Furthermore many users may be familiar with (automated) updates of

anti-malware software. We should not be too surprised at the first outcome above. The underlying cause for the second point may be impossible to fix on a wide scale. The final outcome confirms the Jakobsson's view that relevance is more important than authenticity.

Wright et al. tested and extended the Model of Deception Detection [37] by Grazioli [87] [2]. The researchers aimed to understand how users determine whether the communications they receive are legitimate or not, and claimed that users' web experience and propensity to trust are the two most influential factors in determining whether users will successfully detect deception. In their user study, carefully prepared phishing email messages were sent to participants, and follow up interviews were also conducted to gain insights into the why participants successfully detected deceptions. The participants of this user study were all university students, and the majority of them were well educated in terms of technical knowledge of the Internet. The characteristics of the participants were certainly different from those of the general public. Hence, the findings of this study may not generalise. The model of deception detection will be further discussed in chapter 3.

2.4.1 Timing Attack Techniques

Personalised phishing attacks/spear phishing attacks have much better chances of obtaining victims' sensitive information. To launch such attacks the attacker must first obtain any necessary information such as

[2] Wright's work was published after the completion and publication of the work that form the basis of Chapter 3 of this thesis (The author was unaware of the work of Grazioli.)

victims' names, with whom they have bank account, etc. Bortz et al. have described two timing–attack methods which can be used to obtain private information and have discussed methods for writing web application code that resists these attacks [10].

They call the first method the direct timing attack. An attacker can launch this attack directly from their own machine by analysing the response time from a website. It can expose information such as the validity of a user name at a secured site. In the case of proving validity of a user name at a secured website, they demonstrated the attack method by directly communicating with the web server and carefully timing the response. They use both valid and invalid user names to login to the web server, and compare the time the web server takes to respond to login requests. As shown in Figure 2.6 there is a significant difference between the response times.

The second method is called the cross-site timing attack, which enables a malicious website to obtain information by sending data from a user's computer. The direct attacks are limited to discovery of static information, it can not reveal private information such as a user's current status on Internet e.g. which websites he/she is logged into. In reality, if a user logs into a website, there will be some cache mechanism enabled on the server side or else a cookie will likely be set up on the client side to improve the response time. So if one can make requests as another user, whether a user has logged into a certain website or not by analysing the response time. This attack method is only possible when a user victim is visiting a malicious website. The malicious website contains JavaScript code to make requests to target websites and time the response. Figure 2.7 shows that there are significant timing differences if victims are logged on to the target websites. Using this method, a malicious website, in some

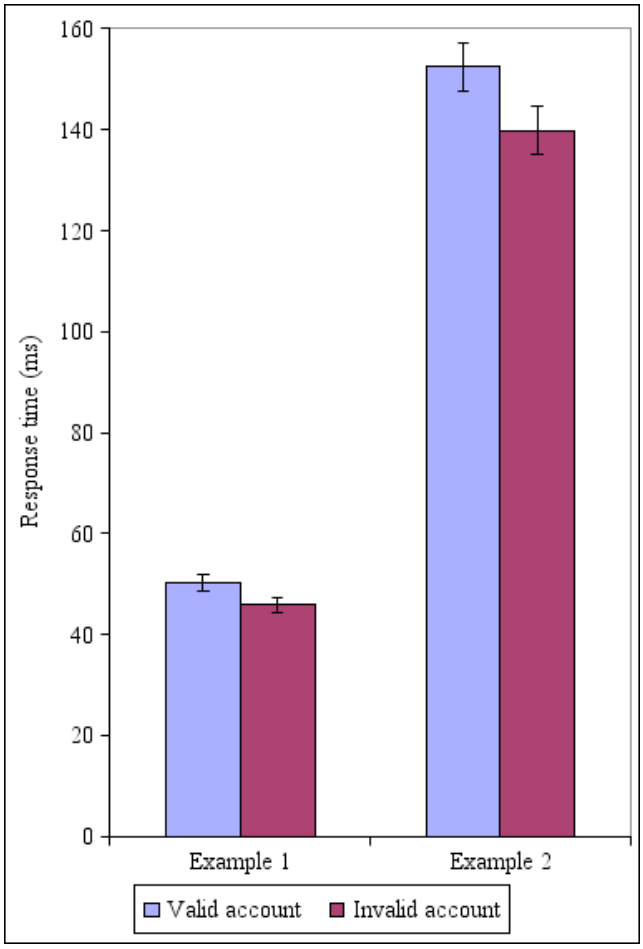


Figure 2.6: The Direct Timing Attack: Response Time Difference [10]

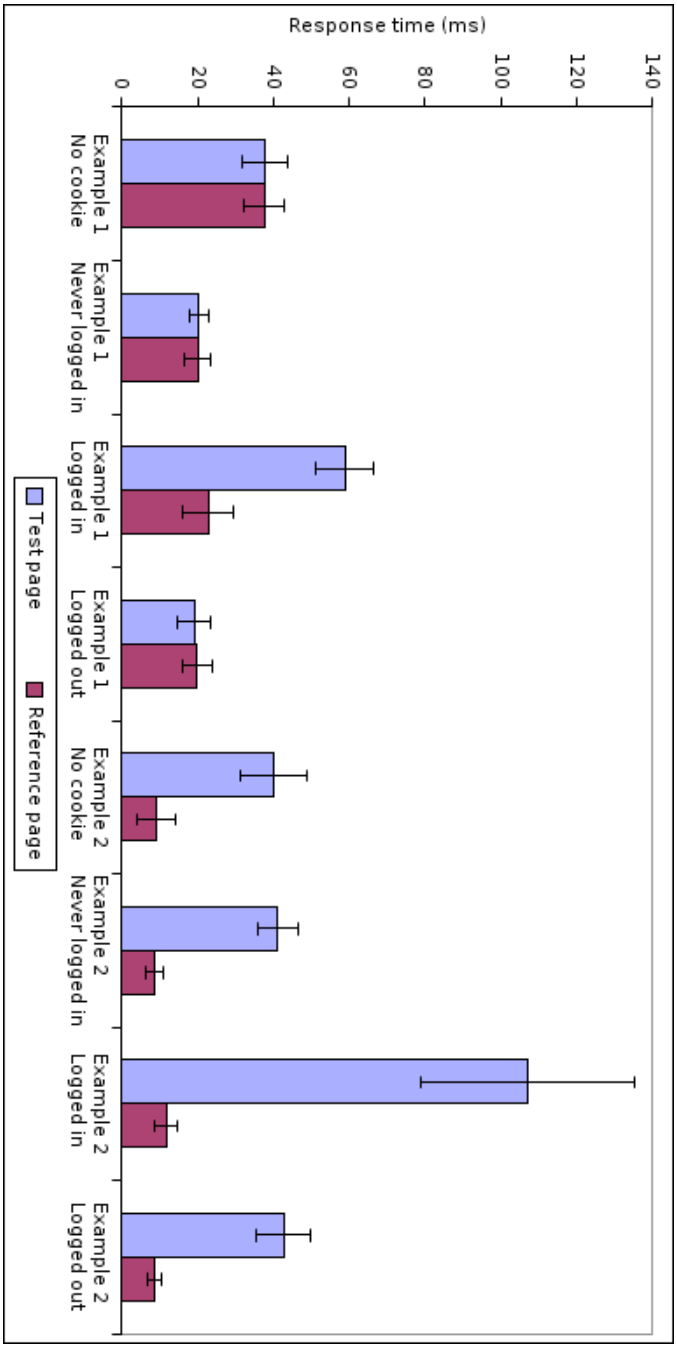


Figure 2.7: The Cross Site Timing Attack: Response Time Difference [10]

cases, could also discover the number of objects in the user's shopping cart. This method could be used to assist in-session phishing attacks or discover users' relationships with certain websites.

Timing channels have been known for some time. For example, the US Department of Defence's principal Trusted System Evaluation Criteria identifies such covert channels as a threat [4]. The exploitation of timing properties led to a very high profile key discovery attack by Kocher [56]. It is perhaps not surprising that social engineering now avails itself of such mechanisms

2.4.2 Discovering Browsing History

A user's browsing history contains information that may allow sophisticated personalised phishing attacks to be created. It could reveal whose customer the user is and who their email account provider is. With this information attackers can tailor their phishing attacks rather than send out random cold phishing messages to a large number of email addresses.

Below, a method that does not exploit any browser vulnerabilities is described. This method takes advantage of a feature of the web browser standard – for a visited URL address the hypertext link pointing to such an address is displayed in a different colour to those hypertext links pointing to addresses that have not been visited. By retrieving the colour property of the hypertext link, attackers can find out whether a user has visited the address the hypertext link points to. Attackers can have those hypertext link embedded into websites under their control, and make

the hypertext link point to the websites which their attacks are targeting. There are two methods to retrieve the colour property: one is by using JavaScript and the other is by using CSS. Sample code is shown below:

```
// Javascript example:

var node = document.createElement("a");

a.href = url;

var color = getComputedStyle(node, null).getPropertyValue.("color");

if (color == "rgb(0,0,255)") {    .....}



// CSS example:

<style>a:visited
                    {background: url (track.php?bank.com);}
</style>
<a href= "http :// bank.com"> hi</a>
```

2.4.3 Retrieving Personal Information in Web 2.0

Personal information is often openly published on websites and public databases. Such information could be used by attackers to launch

spear-phishing attacks and in the worst cases they can directly obtain authentication credentials.

A user's mother's maiden name is often used by financial services as one of the user authentication credentials. Virgil Griffith et al. have invented novel techniques to automatically infer mother's maiden names from public records [38]. As a proof of concept, they applied their techniques to publicly available records from the state of Texas. Other personal information such as date of birth, name, and even home address could also be obtained by scanning social network websites. These techniques, once understood, do not require any insider information or particular skills to implement. They pose serious threats to the integrity of users' authentication credentials.

Social media presents significant opportunities for the unwary to reveal far too much information. In May 2011, Twitter users responded to requests to form their "Royal Wedding Name". Suggestions for how to do this included: start with "lord" or "lady" and forming a double barrelled surname involving the street where you live. Other advice suggested using your mother's maiden name. It may be fun. It is also clearly dangerous from a security point of view. Anti-malware and security specialist Sophos have posted information to indicate why participating in such activities is a bad idea.

2.5 Technology Countermeasures

To counter the phishing threat systematically, new systems and software have been invented. Based on how these systems prevent users from falling victim, I classify them into three approaches: secure authentication methods, indicators and visual cues to help users detect phishing attacks, and phishing attack detection. Users fail to distinguish phishing websites from legitimate websites because of the poor usability of the server to user authentication. The first two approaches focus on improving that, and the third approach tries to reduce the chances of users being tricked by detecting the phishing websites or emails automatically. I will review the major contributions in each approach in this section.

2.5.1 Novel Indicators and Visual Cues

Conventional SSL/TSL digital certificates have failed to provide sufficient secure server to user authentication, because users either do not pay attention to them or they are not able to use the certificates to distinguish phishing websites from legitimate websites. Extended validation certificates (EVs) [43] have been proposed to provide more secure and usable server to user authentication. EVs require more extensive investigation of the requesting entity by the certificate authority before being issued, so an attacker is unlikely to get one. In contrast almost anybody can get a SSL/TSL certificate. In supporting web browsers (most existing browsers support EV, for example Microsoft Internet Explorer 8, Mozilla Firefox 3, Safari 3.2, Opera 9.5, and Google Chrome), more information will be displayed for EV certificates than ordinary SSL certificates. To get an EV

certificate the applicant must pass all criteria set by the Certificate Authority. For every EV certificate issued, the Certificate Authority assigns a specific EV identifier, which is registered with the browser vendors who support EV. Only validated EV certificates receive enhanced display. However, the effectiveness of such certificates is in doubt; a user study has also found that EV does not improve users' ability to recognize phishing websites [45]. The small size of this study's sample base (nine test subjects per cell) is not big enough to strongly support the claim.

Microsoft InfoCard [12] is an identity meta-system that allows users to manage their digital identities from various identity providers and employ them in the different contexts where they are accepted to access online services. Users first register at the identity providers to receive various virtual cards from them. When they go to a web site and are asked for identity data, they click the corresponding virtual card, which will in turn start an authentication process between the current site and the identity provider who issues that card. Again, users do not need to type any sensitive data at all. The major problem with InfoCard is that it needs the web sites and the identity providers who support it to add new functionalities. Since InfoCard is a new way for users to provide their identity information, web sites have to be modified to accept the InfoCard submission, by adding an HTML OBJECT tag that triggers the InfoCard process at the user's browser. The sites also have to add back end functionality to process the credentials generated from different identity providers. Moreover, since InfoCard is an identity meta-system, it needs support from various identity providers, including banks that issue bank accounts, credit card companies that issue credit cards, and government agencies that issue government IDs. These identity providers also need to add functionality to process the InfoCard requests. In order to use InfoCard, users have to contact different identity providers to

obtain InfoCards from them, which introduces an out-of-band enrolment process between the users and the identity providers.

Web Wallet [89] tries to create a unified interface for authentication. It scans web pages for the login form. If such a form exists, then it asks the user to explicitly indicate his/her intended site to login. If the user's intention matches the current site, it automatically fills the relevant web page input fields. Otherwise a warning will be presented to the user. Figure 2.8 illustrates the web wallet in IE 7. The idea of Web Wallet is quite simple and should work if it can accurately detect the login form and prevent users from directly inputting the authentication credentials to a website. This will not be an issue for legitimate websites. However, the same is not true for phishing websites, which may implement their web pages in a way that can evade the login form detection. With the help of JavaScript or Flash this is very easy to do.

2.5.2 Secure Authentication

Personalised information has been used at authentication web pages to improve the usability of the server–to–user authentication. The logic behind this approach is that only a legitimate website should know the personalised information, hence a phishing website cannot do the same and users will be able to distinguish the legitimate and phishing websites. Visual information is mostly used in this approach, for example Yahoo and Bank of America display a picture, which the user has configured previously, at the login web page. Users must pay attention to check the existence of the personalised picture, if no picture is displayed or the displayed picture is wrong then the current website is likely to be a phishing

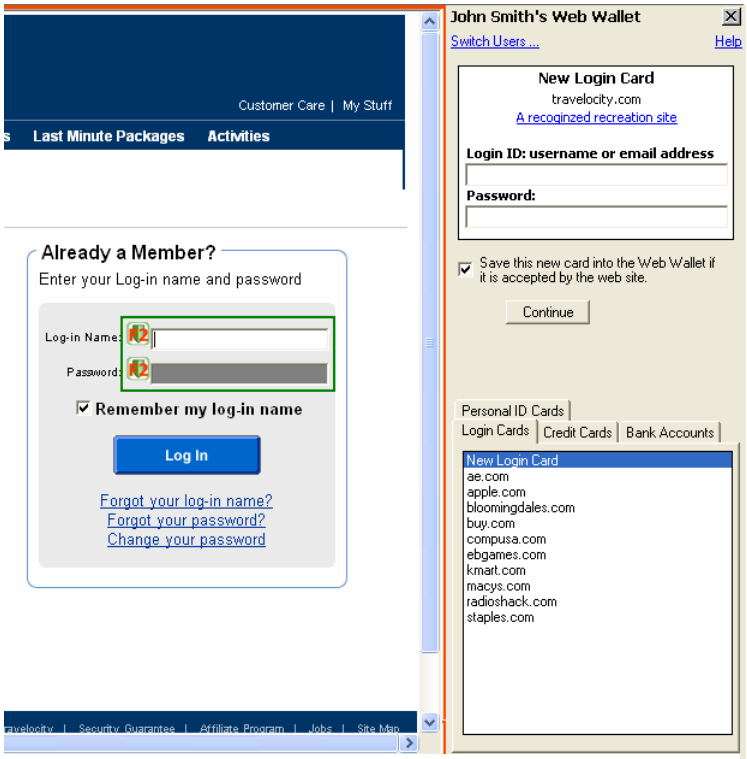


Figure 2.8: Web Wallet

website. Dynamic Security Skins [22] proposes to use a personal image to identify the user's master login window and a randomly generated visual hash to customize the browser window or web form elements to indicate successfully authenticated sites. The advantage of this type of approach over other security indicators is that the message display is much harder to spoof and is at the centre of the user's attention so that it cannot be easily ignored. The potential problem with this approach is that authentication is still not forced and it places the burden on users to notice the visual differences between a good site or interface and a phishing one and then correctly infer that a phishing attack is under way.

Security solution providers such as RSA provide hardware token based multi-factor authentication to prevent phishing attacks. These systems do not try to stop attackers from getting authentication credentials used in the current session, because they use different authentication credentials in future sessions. As long as the hardware token, which is used to generate the valid and unused authentication credentials, is uncompromised attackers can not access users' accounts. USB keys and smart cards are often used as hardware tokens. However, there are some usability issues with this method. It is not cheap to deploy such systems on a large scale. It also requires users to carry extra hardware with them, and these hardware tokens may be damaged or lost. Another uses multi-factor authentication approaches with "someone you know" as a factor [11].

Mobile devices such as personal mobile phones have been used to create more usable multi-factor authentication systems. Software can be installed on the mobile phone to generate a one-time password, and work exactly like other hardware tokens. In addition, since a mobile phone also has a communication capability, the one-time password can be sent to the

phone via SMS text instead of computed on the local device. No software needs to be installed. The disadvantage of this method is mainly the communication cost.

2.5.3 Detecting Phishing Attacks

The most widely deployed type of phishing detection system is the blacklist based phishing website detection toolbar. Almost every web browser has this feature as default, e.g Internet Explorer [63], Firefox 2 with google's safe browsing [67]. There are also third party toolbars such as Netcraft toolbar [68] and eBay toolbar[44], etc. These systems check whether the URL of the current web page matches any URL in a list of identified phishing web sites. The effectiveness of the detection depends on how complete the blacklist is and how timely the blacklist is updated. This type of detection system has a very low false positive rate; however, there is inevitably a time gap between the launch of a phishing attack and the URL of the phishing website being added to the blacklist. During the time gap, users are at their most vulnerable as there is no protection at all. According to an anti-phishing toolbar evaluation study [90], IE7 is the best performing phishing detection toolbar, but it still missed 25% of the APWG phishing URLs and 32% of the phishtank.com phishing URLs.

SpoofGuard[16] is a typical signature and rule based detection system. It first examines the current domain name and URL with the intention to detect phishing websites that deliberately use the domain names or URLs similar to those of the targeted sites. Secondly, the content of the web page such as password fields, embedded links, and images, are analysed. For images, it will check whether identical images have

been found on other sites the user has visited. If it does, then it is possible that the fraudulent site copied the image from the legitimate site. Finally, spoofGuard computes a score for each web page in the form of a weighted sum of the results of each set of heuristics. There are three possible outcomes:

- if the score is higher than a certain threshold, the toolbar warns the user that the current website is fraudulent and displays a red icon;
- if the score is lower than the threshold but there are heuristics triggers, the toolbar displays a yellow icon which indicates that it cannot make a determination about the site.
- if no heuristics are triggered, a green icon is displayed.

The weights for each set of heuristics can be modified by users as well. SpoofGuard runs on Microsoft Windows 98/NT/2000XP with Internet Explorer. Despite its relatively high detection rate, it also suffers high false positives [90]. In addition, most users are unlikely to be able to adjust the weights of each set of heuristics, and as a result the detection accuracy may be reduced.

CANTINA [91] detects phishing websites based on the TF-IDF information retrieval algorithm. Once a web page is loaded, it retrieves the key words of the current web page and five key words with the highest TF-IDF weights are fed to the search engine (in this case it is Google). If the domain name of the current web page matches the domain name of the 30 top search results, it is considered to be a legitimate web site, otherwise, it is deemed to be a phishing site. (The value of 30 has been

determined by experiment to provide a balance between low false positive and high detection accuracy.) According to the developer's evaluation, it can detect over 95% of phishing websites. However, this does not necessarily mean in reality 95% of phishing will be detected. Once the method of the detection is known to attackers, then they can easily bypass the detection by using images instead of text, or using Javascripts to hide the text, or using keyword stuffing to mislead the TF-IDF algorithm.

Ying Pan et al. have invented a phishing website detection system which examines anomalies in web pages, in particular, the discrepancy between a web site's identity and its structural features and HTTP transactions [72].

Many anti-phishing email filters have been invented to fight phishing via email, as it is the primary channel for phishers to reach victims. SpamAssassin [3], PILFER [27], and Spamato [7] are typical examples of those systems. They apply predefined rules and characteristics often found in phishing emails to analyse incoming emails. PHONEY [14] is different from the phishing email detection system mentioned before. It tries to detect phishing emails by mimicking user responses and providing fake information to suspicious web sites that request critical information. The web sites' responses are forwarded to the decision engine for further analysis. However, its ability to decide what type of information is actually being requested is limited and the approach is easily bypassed. (An attacker can relay the user input to the legitimate website, so that they could behave exactly like the legitimate website.)

2.5.4 Phishing Attacks Threat Modelling

Although phishing attacks have caused serious financial damage and have reduced users' confidence in the security of e-commerce, there is still a lack of methods to systematically analyse a given user authentication system for both system and user side vulnerabilities. As far as I am aware, the only published work that analyses the usability vulnerabilities of a system is by Josang et al. [53].

Josang et al. present four "security action usability principles":

1. Users must understand which security actions are required of them.

2. Users must have sufficient knowledge and the ability to take the correct security action.

3. The mental and physical load of a security action must be tolerable.

4. The mental and physical load of making repeated security actions for any practical number of instances must be tolerable.

These are supplemented by four "Security Conclusion" principles:

1. Users must understand the security conclusion (e.g the owner of a digital certificate, or the top level domain of a URL) that is required for making an informed decision.

2. The system must provide the user with sufficient information for deriving the security conclusion.

3. The mental load of deriving the security conclusion must be tolerable.

4. The mental load of deriving security conclusions for any practical number of instances must be tolerable.

The research described in the paper described the vulnerabilities caused by violating the proposed usability principles. However, the vulnerabilities they identified are not comprehensive enough for practical uses. They have not considered all usability principles, for example, that the alert given to user at the interface should be active enough to grab their attention.

They also suggested that such vulnerabilities should be considered at the same time as analysts consider technical vulnerabilities. This suggestion makes sense as integration with existing methods should increase the usability of the threat modelling method itself. However, the authors have not introduced a clear and systematic approach that one could follow. As illustrated in the three case studies; the threat modelling process lacks structure and relies very much on one's own experience and judgement.

2.6 Limitations of Current Work

The foundation for developing effective protection against phishing is an understanding of why phishing attacks work. Extant user studies have shed light on many important aspects of this issue. They have addressed how users behave in given situations, what are their habits,

and how they perceive trust and security in cyber space. However, these studies have not been guided by a coherent framework. As a result, the knowledge discovered by these studies varies in depth and lacks structure for practitioners to absorb and use as design guidance. These studies are also too closely linked to current technology; findings could become invalid very quickly as systems, technologies, and user characteristics and behaviours evolve.

There is also little literature addressing the threat modelling of the user side of a system. Users are now frequently targeted by attackers, and to be able to systematically discover the vulnerabilities posed by users is just as important as discovering system vulnerabilities. It would be very useful to be able to do the threat modelling on the user side of the system at the design stage.

Finally, existing detection systems all have strengths in particular areas; no system would appear superior in all aspects. Most of these detection systems react to what phishing attackers have done, and the detection algorithms are based on the manifestation of discovered phishing attacks. As a result, attackers can evade detection by changing their tactics.

To address the issues identified above, in the next chapter I introduce a phishing–user–interaction model to put existing knowledge into a coherent framework, and identify the fundamental reasons why users fall victim to phishing attacks. Chapter 4 presents new techniques for identifying user-based threats in web authentication systems. Drawing on the few principles established in the literature (e.g. these by Josang et al.) but also providing significant extension. Chapter 5 presents a proof-of-concept toolset that aims to overcome some of the disadvantages posed by anti-phishing tools. Finally, Chapter 6 evaluates and concludes

the work done and discusses future research areas.

Chapter 3

A Phishing-User Interaction Model

This chapter introduces a psychological model to capture the general process of the decision making during user-phishing interactions and identifies important factors that can influence the outcome of such decision making. At the end of the chapter I also show how this model can be used to guide a wide range of applications:

- designing security tools/indicators;
- evaluating how well a phishing detection tool can assist users to detect phishing attacks; and
- designing effective and efficient user education methods.

3.1 Study Approach

The decision making process during user-phishing interaction was analysed using a collection of attack incidents. The analysis method is cognitive walkthrough [13].

3.1.1 Why Decision Making?

The goals of phishing attacks are achieved by causing victims to carry out actions which lead to the compromise of confidential information. Since the actions users take are the realization of the decisions they have made, attacks try to manipulate victims to make certain decisions. Hence, how people make decisions when encountering phishing attacks is what really matters.

3.1.2 Attack Incidents

A significant collection of social engineering and phishing attacks were analysed with a view to extracting a behavioural model for the user. Social engineering attack incidents are used to understand the big picture about how people are manipulated into making decisions that would satisfy an attacker's goals. Phishing examples provide a particular focus for our consideration.

The attack incidents collected are all real; 45 social engineering attacks and 400 phishing attacks (collected from APWG [73] and Millersmiles

[2]) are analysed. These attack incidents cover a comprehensive range of attacking techniques, attack delivery channels, vulnerabilities exploited, and attack goals. Unlike phishing attacks, there is no publically accessible social engineering attack incident data set. I collected those social engineering attacks by editing the attack incidents reported by many security practitioners working in the field. Besides the description of attack incidents, a basic analysis for each attack can also be accessed at http://www-users.cs.york.ac.uk/~xundong/se/dse.html. This data set is the first of its kind, it can be useful for future research in this field as well as security education. An example is shown in Figure 3.1.

3.1.3 Methodology

Interviewing victims of attacks provides one means of gaining insight into their decision making process. However, locating victims is not easy[1] and there would be difficulty ensuring that those interviewed really were representative in any meaningful way. Victims are often embarrassed at being conned, and reluctant to come forward in the first place. Another approach would be to recreate such attacks in a controlled user study. However, the range of attacks is significant. Recreating them under experimental conditions would be very expensive. In addition, the context in which some of these attacks took place has great influence on how victims behave, and it is also very difficult to recreate such contexts.

A plausible approach to studying the collected attacks is the cognitive walkthrough method [13]. This is an inspection method often used to assess and improve usability of a piece of software or a web site. It can

[1] attack reports do not reveal victims' identities

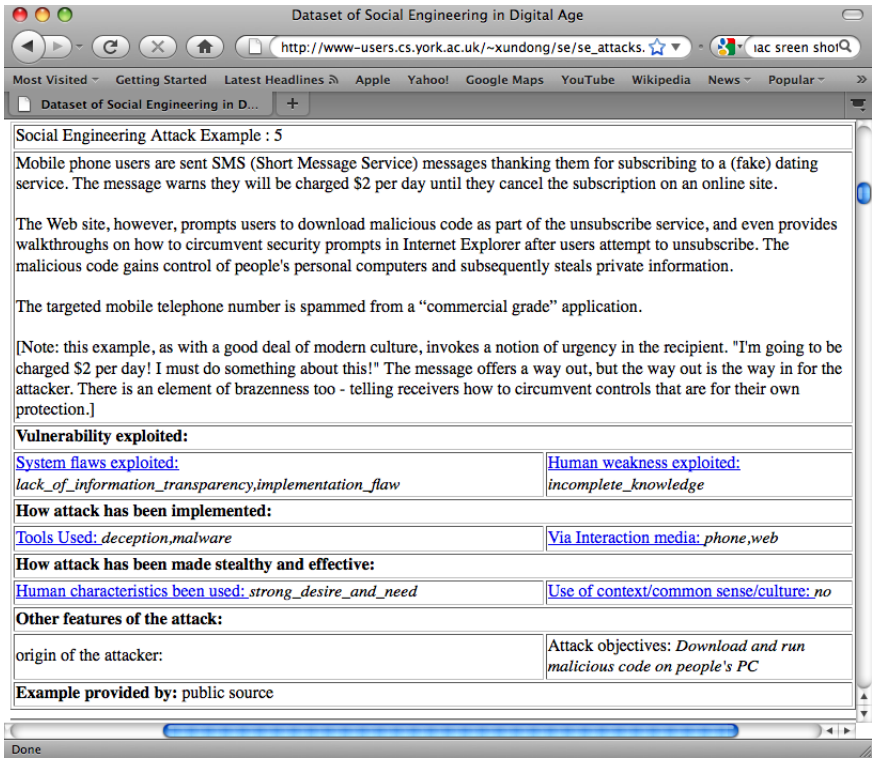


Figure 3.1: A Social Engineering Attack Incident Retrieved From The Collected Data Set

generate results quickly with low cost.

The cognitive walkthrough focuses on the tasks users need to perform and on the user interface through which users complete their tasks. There are five steps in the procedure of the cognitive walkthrough [13]:

- Define the inputs to the walkthrough;
- Convene the analysis;
- Walk through the action sequences for each task;
- Record critical information;
- Revise the interface to fix the problems.

The final step is not required for the purpose of this research.

The first stage usually requires the definition of the following four factors [13]:

- Identification of the users. Cognitive walkthrough does not usually involve real users, so this is to identify the characteristics of the targeted user group. Here the targeted users are the general public who are capable of using the software but do not possess much knowledge of the underlying technology.
- Sample tasks for evaluation. Here these are the attack incidents described in the previous section.

- Description (mock-ups) or implementation of the interface. In this case, the user interfaces considered are not prototypes but the ones that are most used by the general public already. For example, the web browsers considered are IE7 and Firefox.

- Action sequences (scenarios) for completing the tasks. These are the actions sequences exhibited when users fall victim to the attacks.

For each attack incident, having identified the action sequence the analysis is done by first asking a series of questions for each step. Below are the questions to be asked.

- What is the context of the user-phishing attack interaction?

- What are the users assumptions and expectations?

- What is the false perception attackers try to engineer in the victim's mind?

- What would be the expected feedback from the user interface to the victim for each action they have taken?

- What information **obviously** available at the user interface can be selected/interpreted to form false perceptions?

- What information **obviously** available at the user interface can be selected/interpreted to form an accurate perception?

- Would users know what to do if they wished to check out the authenticity of whom/what they are interacting with? If so, is it

easy to perform the check consistently?

The answers from these questions provide insights into how users interact with a phishing attack and form the basis for analysing why they fall victim. Based on these analyses, a user-phishing interaction model is abstracted. The findings described in the following sections are all based on the analysis conducted during the walkthroughs. The model starts when users encounter a phishing attack and finishes at the point where all actions have been taken. An example of the cognitive walkthrough can be found in the appendix.

3.2 Overview of the Interaction

During a human-computer interaction, the user's role in phishing attacks is to retrieve relevant information, translate the information into a series of actions, and then carry out those actions. The overview of the model is shown in Figure 3.2.

There are three obvious types of information users can use when encountering phishing attacks, and they are connected to users in Figure 3.2 by solid arrows. External information is information retrieved from the user interface (including the phishing emails/communication) as well as other sources (such as experts' advice). The context is the social context the user is currently in. It is the user's perception of the state of the world, comprising information on things such as recent news, what is happening around the user, the user's past behaviour, social networks, etc. Knowledge and context are built up over time and precede the phishing

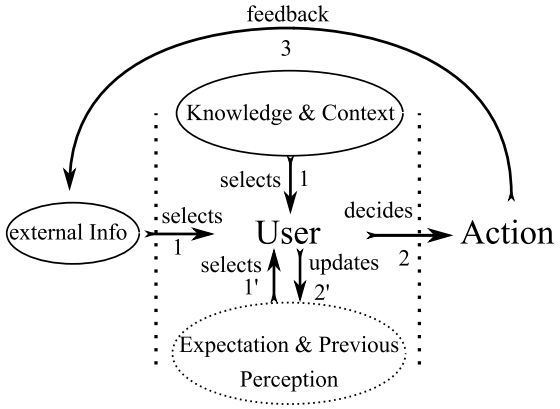


Figure 3.2: The Overview of User-Phishing Interaction

interaction. External information must be specifically retrieved during the interaction. The information items displayed on the user interface are the most obvious and immediately available external information to users. As a result, they are always used in the interaction. External information from other sources is selected only when certain conditions occur, such as a user becoming suspicious.

It usually takes more than one step to reach the action that could lead to the disclosure of the information that attackers seek. For example, an email based phishing attack may require victims to read the phishing emails, click on embedded URL links and finally give out confidential information at the linked phishing web sites. For each completed action, users have expectations of what will happen next (the feedback from the system) based on their action and understanding of the system. This expectation, together with the perception constructed in the previous

steps, are carried forward when users decide whether to take the next action. Since these two types of information exist only during the lifetime of the interaction, we present them differently in Figure 3.2. This decision making happens each time before an action is taken. Thus Figure 3.2 includes a feedback arrow from action to external info. Among the information that could influence one's decision making, the attackers can directly affect only the information displayed by the user interface.

3.3 The Decision Making Model

From analysis of the walkthroughs two types of decisions that users make during user-phishing interactions are apparent:

1. planning what actions to take; and

2. deciding whether to take the next planned action or not.

The decision regarding what actions to take happens before the second type of decision making. The decision regarding what actions to take will be referred to as the primary decision, and the other will be referred to as the secondary decision. The secondary decision clearly takes place after the primary decision. It seems the expectation and perception constructed in the primary decision making process influences significantly what, and the amount of, external information a user selects during the secondary decision making process. As a result, users' ability in discovering false perceptions during the two types of decision making are not the same. The fundamental steps within the two decision making process are

identical, although the detailed characteristics of each step are different (the differences as well as their implications are discussed in later sections). With reference to general decision making theory [9, 51, 55, 78, 75], both types of decision making can be divided into the following three stages:

- construction of the perception of the current situation;
- generation of possible actions to respond; and
- generation of assessment criteria and choosing an action accordingly.

These are now discussed.

Construction of the Perception

The perception that users have constructed in this stage describes the situation users are facing and goals they want to achieve. The perception is constructed by first selecting the information, and then interpreting the information selected. The author suggests using the following four aspects to describe perception:

- Space
  - Direction of the interaction, e.g. who initiates the communication.

  - Media, through which the communication and interaction occur, for example, via email, phone, etc.;

- Participants

  - Agent, who begins the interaction;

  - Beneficiary, who benefits;

  - Instrument, who helps accomplish the action;

  - Object that is involved in the interaction, for example, it could be personal/business confidential information that the attacker would like to obtain;

  - Recipient, who is the party that an agent tries to interact with;

- Causality

  - Cause of the interaction, what caused the interaction to occur;

  - Purpose of the interaction, what an agent tries to achieve;

- Suggestions

  - Explicit suggested actions to reach the goal; and

  - Implied actions to reach the goal.

In the rest of this chapter these four aspects will be used to describe a user's perception.

The mismatch between a user's perception and actual fact has previously been described as the mismatch between a user's mental model of the information system and actual implementation [31]. This understanding is too general to be useful in detailed analysis. Using the above four aspects we can discover that the false perception, which phishers try to engineer, has two mismatches with actual fact:

1. some perceived participants are not actual participants; and

2. the perceived causality (the cause of the communication and the consequence of the suggested actions) is not the actual causality.

These two mismatches exist in every phishing attack analysed, because the real participants are the phishers, their phishing websites, etc. rather than the legitimate organisations or persons whom the victims trust; and the true motive is to steal people's confidential information rather than any causality phishers suggest. Failure to discover such mismatches allows phishing attacks to succeed. A later section of this chapter discusses how users could discover such mismatches and why they often fail to do so.

During primary decision making, a perception (defining the four aspects of the perception) is created from scratch. Once this initial perception is created, during secondary decision making users focus on refining and confirming the perception created by selecting information from feedback from the actions they have already taken. There is a significant amount of psychological research [75] to suggest that when selecting information from feedback, users have a strong information selection bias – they tend

to select information that confirms their initial perception and ignore facts that contradict it. Users are unlikely to change their perception unless they discover facts or information that contradict to the existing perception.

Generation of Possible Solutions

Factors such as time, knowledge, available resource, personality, capability, etc. all affect the set of actions one can generate. Interestingly, the author's analysis of collected phishing attacks suggests that the user's intelligence in generating possible actions is not one of the major factors in deciding whether they fall victim to phishing attacks or not. It is not people's stupidity that makes them fall victim to phishing attacks.

Analysis of the set of phishing attacks collected by the author revealed an interesting feature: the victim generally does not need to work out a solution to problems presented. Rather, the attacker kindly *provides* the victims with a "solution", which is also the action they want victims to take. For example, an email message stating that there is a problem with a user's authentication data may also indicate that the problem can be "solved" by the user visiting a linked website to "confirm" his/her data. If the situation is as presented, then the "solution" provided is a rational course of action. Unfortunately, the situation is not as presented.

In some attacks the solution is not explicitly given, but it can be easily worked out by common sense. For example, the attackers first send users an email appearing to be a notice of an e-card sent by a friend. Such e-card websites are database driven, the URL link to access the

card often contains parameters for the database search. The URL link that the attackers present to victims has two parts: the domain name of the website and the parameter to get the card. The former points to the legitimate website, but the parameter is faked. As a result, the user will see an error page automatically generated by the legitimate website. At the end of the email attackers also present a link for people to retrieve the card if the given URL link is not working. This time the link points to the phishing website which will ask for people's address and other personal information. In this attack victims have not been told to click the spoofed URL link, but "common sense" suggests using the backup link provided when the first link they have been given has failed.

Phishing attacks can be viewed as follows. The attacker tries to engineer a false perception within the victim's mind, and also tries to simplify the solution generation stage by telling users what actions he/she should take to respond to the false perception. He or she must now decide only whether to take the suggested means of solving the problem – the user does not feel a need or desire to generate any alternatives. Simplifying users' decision making processes might make users spend less time on selecting information. The chances of users discovering any mismatch of perception is also reduced.

This is one of the important differences between the user-phishing interaction and the general user-computer interaction where users have to work out the actions to reach the goals themselves. In fact, users need to do little to generate any solutions, so the solution generation is not presented in the graphical model illustrated later in this chapter.

Generation of Assessment Criteria and Choosing the Solution

To decide whether to follow the suggested action is the focus of this stage. A user generates the criteria to evaluate the resulting gains and losses of possible actions, then evaluates those actions and chooses the best one. Sometimes the criteria are already established, such as one's world view, and personal preferences. Sometimes criteria must be carefully developed. Each individual's experience, knowledge, personal preferences and even emotional/physical state can affect the assessment criteria. However, most of the phishing attacks analysed did not take advantage of the differences between users, instead they took advantage of what users have in common.

In all phishing attacks analysed, besides engineering a false perception, attackers also suggest a solution to respond to the false perception. The solution suggested is rational according to the false perception and it is also very likely to satisfy some of the victim's assessment criteria. For example, everyone wants their bank account authentication credentials to be secure, and so a solution which appears to increase security "ticks the box" – it simply appears to provide them with something they actually want. We should not be surprised when they accept this solution and act on it. Similarly, users will (usually) feel a need to follow an organisation's policy, and so will most likely follow instructions that appear to come from authority. Being friendly, helpful to others, curious to interesting things, and willingness to engage in reciprocative actions (often used by social engineering attacks), are all common criteria.

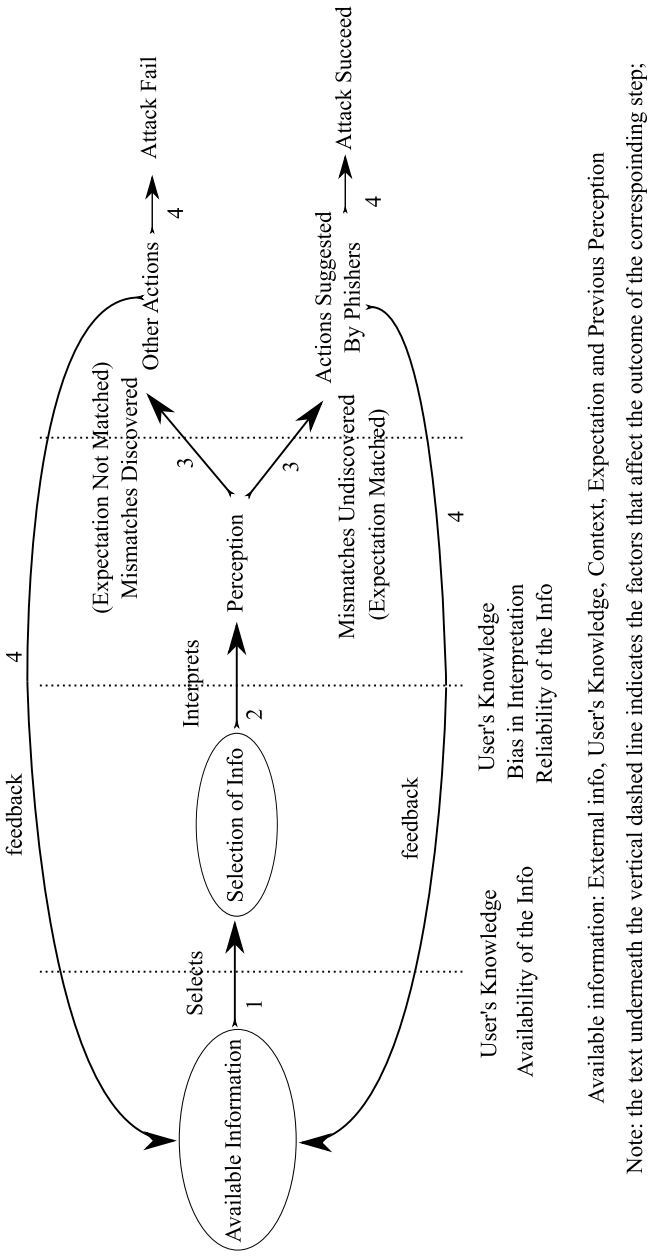
As long as the victims have not discovered the mismatch between the false perception and the truth, they would indeed want to follow the suggested

actions and the evaluation of this stage would most likely be "yes". To make the user's position even worse many users have a natural tendency to follow the instructions from the authorities or their service providers. As a result, they may not even carefully evaluate the suggestions provided at all. Again this stage is also secondary in deciding whether users fall victim to phishing attacks, so I exclude it from my graphical model.

3.3.1 Graphical Model

In a user phishing interaction, the user first decides the sequence of actions (primary decision), and then follows the decision making process concerned with whether to take the next planned action (secondary decision). The secondary decision making is carried out repeatedly prior to each action a user takes. Both types of decision making processes comprise the same three stages. However, their characteristics are different.

In the primary decision making process, there are no perceptions constructed already. Users need to form their initial perceptions by selecting a wide range of external information. In contrast, in the secondary decision making process, users have previously constructed perceptions and expectations to influence their selection of external information. The evaluation of whether to take the chosen solution (the next planned action) mainly depends on whether the feedback from previous actions matches expectation, while the evaluation stage for the primary decision is more general and flexible. For example, when users click the hyper links embedded in phishing emails, they expect the web browser to present to them the legitimate website that the phishers are trying to impersonate. If they perceive the website presented by the browser (the feedback) as



Available information: External info, User's Knowledge, Context, Expectation and Previous Perception
Note: the text underneath the vertical dashed line indicates the factors that affect the outcome of the correspoinding step;

Figure 3.3: The Decision Making Model

the legitimate one (the expectation), then they will carry out the next planned action, which could simply be giving out their authentication credentials. If the users have constructed a correct perception (they have been presented with a phishing website), which does not match their expectation, their next planned action will not be taken.

Regardless of the type of decision making process, construction of an accurate perception is key to detecting phishing attacks. People's ability to work out solutions and evaluate the alternatives plays little part in preventing them from falling victim to phishing attacks in both types of decision making. Because the perception construction is so important, it is also the main focus of our graphical model, which is shown in Figure 3.3.

As there are multiple actions in each interaction and for each action users have a decision to make, the model has two loops. In the model, each cycle represents the decision making for one action and the cycle begins with the "available information" and ends when an action is taken. Once the action is taken then there will be new information available and decision making for the next action begins. For each interaction, the first cycle of this model describes the decision making regarding the planning of the sequence of actions to take, and the rest of the cycles are concerned with deciding whether to take the next planned action or not. The model is validated against the analysed phishing attacks in the cognitive walkthrough.

3.4 False Perceptions and Mismatches

Because perception construction is the most important factor in deciding whether users fall victim to phishing attacks, in this section the perception construction stage is modelled further. An examination is carried out of how those mismatches could be discovered, and of why many users fail to do so.

3.4.1 How Mismatches Can Be Discovered

There are two types of mismatch: one concerns the participant, the other concerns causality. The discovery of the mismatch of participant (refer to section 3.3 for how we define the participant) mainly relies on the information users select from the user interface. There are two types of information presented to users at the interface:

- the metadata of the interaction; and
- the body of the interaction.

The metadata of the interaction is the data which systems use to identify an entity. It can be used to form the space and participant aspects of the perception. Example metadata includes the URL of the website, digital certificates, and the sender's email address. The body of the message is used for conveying the semantic meaning of the interaction. It can be the text content of an email message, webpage visual content, a video or an audio clip.

A participant mismatch can be revealed by discovering the inconsistency between the body of the interaction and the metadata of the interaction. For example the content of the phishing webpage suggests it is Ebay's website while the URL address suggests it is something else. The mismatches could be revealed if users selected both metadata and the body of the interaction to form the perception. Users are good at understanding the body of the interaction, however, many users may not possess sufficient knowledge to understand the metadata of the interaction. Previous user studies [21, 50] have also confirmed users' lack of technical knowledge. To make the task of understanding the metadata of interaction even harder, phishers often uses visual spoofing techniques to make the metadata look like the counterpart of the impersonating targets. For example, as shown in Table 3.1 phishers often use URL link manipulation to make the URLs of the phishing websites look legitimate. As a result, a lack of ability to understand the metadata of the interaction contributes to users' failure to discover participant mismatches.

Moreover, due to vulnerabilities in the system implementation and design, the metadata could be spoofed to appear consistent with the body of the interaction. For example, the sender's email address could be configured to display any email address the attacker wishes; an example is illustrated in [49]. Rachna [21] has provided a list of techniques phishers use to spoof metadata. As a result, to discover the mismatch of the participants, users have to discover the inconsistency between the metadata and low level system data that is not presented to users at the interface. For example, if the attacker faked the sender's address, which has not been associated with a digital certificate, then users need to determine the original SMTP server who sent out this email and compare it with the server from where the email with the same email address is normally sent or investigate the IP address from which the email has been sent. Another example is when

Table 3.1: A Sample of Phishing Websites URLs

1	http://www.hsbc.com-ids-onlineserv-ssl-login-secure-id-user.708210.12secure.com.tw-rock.org/creditcard/
2	http://myonlineaccounts2.abbeynational.co.uk.dllstd.co.uk/CentralFormWeb/Form?action=960963654803377102593693292280715313694104202156 12
3	http://74.94.36.10/www.paypal.com/cgi-bin/webscr=cmd=p/index.php
4	http://www.ebay.cd.co.uk/eBayISAPI.dll?ViewItem&item=150258354848&id=6
5	https://www.longin.com/Login?continue=http://www.google.com/&hl=en

the phishing website's URL address is spoofed[2] to be identical with the legitimate site. Unless users can analyse the IP address of the phishing website as well as the IP address of the legitimate website, users cannot discover the mismatch. Other examples also include caller ID spoofing by using VoIP systems, which allow users to specify the outgoing call number.

User education [73, 17, 18, 25, 30, 57, 62] has been used as a means to protect users from phishing attacks. But to discover the mismatches when meta-data is spoofed requires extra tools and knowledge which it would seem unrealistic to expect many users to have. Such required knowledge cannot be passed on to users without significant financial investment and time. It is the system designer's responsibility to ensure information

[2]A phishing attack that targets Citi Bank customers has used this technique. The story is published at http://blog.washingtonpost.com

displayed on the user interface is resistant enough against most spoofing attacks, especially the meta-data. Furthermore, if designers of security tools and indicators do not ensure their meta-data is reliable against spoofing attacks, these tools may provide new avenues for phishers to engineer more convincing false perceptions.

To discover mismatches of causality is difficult. Such mismatches will be discovered if users are aware of certain contextual knowledge that contradicts the story described in the body of the interaction. For the phishing attack example illustrated in the Chapter 1, to discover the mismatch the RBS customer needs to know how Lloyds TSB banking group handle the existing bank accounts of HBOS customers or they need to know Lloyds TSB would never contact them by email regarding such an important issue. However, it is very unlikely many users have such knowledge, especially users who have never been customers of Lloyds TSB before. This phishing attack is sophisticated also because there is considerable truth in its email. The truth in the phishing email is likely to reduce the suspicion of users. There is another phishing attack for VISA card holders. To make the online credit/debit card payment more secure VISA has launched a "verified by visa" scheme. In this scheme users create personal messages and passwords for their cards, and when they pay for online purchases, users will be asked to provide the password as well as the card details. Phishers have taken advantage of this scheme and send users phishing emails which ask them to join this scheme if they have not done so, although links provided in emails lead to phishing websites. In this case, it would be very difficult to discover the mismatch of causality unless users are aware of when VISA will send emails to its users and what emails have been sent. Unlike a participant mismatch, a mismatch of causality is not always discoverable from the user side, as users can not be expected to possess the required contextual knowledge.

3.4.2 Why Users Form False Perceptions and Fail to Discover Mismatches

Users do not solve the actual problem nor respond to the actual situation, they make decisions purely based on their perception [75]. Unsurprisingly, in phishing attacks victims invariably perceive the situation erroneously and solve the wrong problem. The victim's response is flawlessly rational according to the perception; he/she may execute an entirely cogent plan to react to the perceived situation. The problem is, of course, that this underpinning perception is simply wrong.

To answer the question why users form a false perception and fail to discover the mismatches, researchers have tried to observe how users behave in controlled user studies [21]. I feel this question can be answered more completely and accurately by referring to the important steps illustrated in the graphical model of user-phishing interaction. The reasons why users fail to execute each step correctly can provide answers to this question, and they are:

- The selection of the information (especially metadata) is not sufficient to construct an accurate perception to reveal mismatches of participants;

- The information selected has not been interpreted correctly.

- Once engaged in the secondary decision making (regarding whether to take the next planned action) users' ability to construct an accurate perception might drop because of early inaccurate/incomplete expectations and less critical thinking generally.

Insufficient Information

There are five causes for insufficient selection of information:

- metadata presented to users at the user interface is not sufficient;
- metadata presented at the user interface is vulnerable for spoofing attacks;
- Users do not have the knowledge to allow them to select sufficient information;
- Users have not paid enough attention to security and hence some important information has not been selected;
- It is physically or mentally intolerable to select sufficient information consistently.

Some user interfaces do not provide enough metadata information. For example, in mobile phone communication, the caller's ID can be hidden. In the UK, banks or other organisations often call their customers without displaying the phone number. The phishers can just call a customer to impersonate legitimate organisations by hiding their phone number as well. Even when the phone number is displayed, the users may still not be sure of the identity of the caller, because there is still no information (at the user interface) that can help them to decide who actually owns a number. Here recognition of voice would not help because most likely the user will not have built up a relationship with any specific caller from that organisation (e.g a legitimate caller could be one of a considerable

number in a call centre.)

If the metadata presented at the user interface level can be spoofed, then users would be forced to select extra information to verify the metadata displayed at the user interface. Often such additional information is not provided to users at the interface. To access the extra information requires more knowledge and skills which many users do not possess. The sender's email address is a prime example. That email address can be set to any text string, and yet many users only select it to decide who sent the email.

It is not only technical knowledge users are lacking. Contextual knowledge is also required in some cases. Some organisations and companies share users' information, and users can login to their accounts by using any member's websites. For example, these UK's mobile phone retail companies (CarPhoneWarehouse, E2save.com and Talktalk.com) share their customer information, and users can use the same authentication credentials to access their accounts at any one of the three websites. But how could a user tell whether a site belongs to a legitimate organisation or a phishing website by looking at the URL alone?

Users do not pay enough attention to security related meta-data; often it is simply ignored. This may be because they are more interested in productivity; they want to solve the problem and react to the situation as efficiently as possible and security related metadata seem unrelated to many users' primary goals.

Misinterpretation

The way people interpret the information selected is not error-free. I have summarised biases within users' interpretation of information [48, 50]:

- The existence of 'HTTPS' in the URL means the site is not a phishing website and may be trusted. (HTTPS only indicates the use of TLS/SSL protocol, phishing websites can use it too.)

- The appearance of the padlock at the bottom of the browser or in the URL bar means that the web site visited is not a phishing website and should be trusted. (The padlock only indicates secure communication and it has nothing to do with the identity of the website.)

- The appearance of the digital certificate means the site is secure and should be trusted. (The phishing website can have a digital certificate as well, it is the content of the digital certificates that matters not the appearance.)

- The sender's email address is trustworthy. (In reality it may be spoofed.)

- The professional feeling of the page or email means it might come from the legitimate source.

- Personalisation indicates security.

- Two URL's whose semantic meanings are identical link to the same

websites, for example www.mybank.com equals www.my-bank.com.

- The text of hyper link indicates the destination of the link.
- Emails are very phishy, web pages a bit, phone calls are not.

Besides the misinterpretation, some users also may not be able to interpret the information presented to them at all. This is often the case for some security warnings or error codes. If people do not understand the information they receive, they will not be able to use it to form any meaningful perception and discover the mismatches.

Drop in Perception Ability

In general, humans have been discovered to have information selection bias during decision making. That is many people tend to select only the information that confirms their existing perceptions and beliefs, and they ignore information that contradicts them [41, 61, 75, 85]. Phishing has taken advantage of this aspect of human nature, although this exploitation may not be deliberate.

Phishing attacks first reach users via communication channels such as email. It is at this stage where users will have to establish their initial perception of the interaction. If a false perception is created, then it could trigger information selection bias for the secondary decision making. As a result, users are much less likely to pay attention to the information passively presented by the meta-data and other security indicators. The stronger the false perception is, the bigger the information selection

bias could be. This human factor can very well explain why spear phishing is more effective than just a random phishing email, as the personalised information in spear phishing enhances the likelihood of a false perception significantly. This human factor can also explain the result of the user study by Serge Egelman et al [26], in which researchers found that participants pay little attention to passive warnings on web browsers, and only active warning messages have an effect if participants have already constructed a false perception after reading phishing emails.

The drop in perception ability because of information selection bias in secondary decision making is an important insight, and the understanding of this aspect of human nature can be applied to help users discover the mismatch of participants. The details will be discussed in the next section.

3.5 Suggestions and Guidelines

3.5.1 Security Tools/Indicators Design

Security Indicators Should Be Put In the Right Place and At the Right Time

In the interaction model the first step of decision making in each cycle is to select available information, which will be the base for creating the perception and deciding what action to take. So to provide users with the right information in the right place and right time is vital to protect

users from falling victim to phishing attacks.

There is little evidence to suggest that the difference between the two types of decision making processes has been well understood by system designers and security professionals. Little research has focused on providing usable security indicators in email clients and other forms of communication where the primary decision making takes place. Instead, the focus has been mainly on improving the usability of authentication on web pages where users decide whether to take the next planned action. However, in secondary decision making, users' ability to form accurate perceptions is compromised by information selection bias. This may explain why limited success has been achieved despite a number of tools and indicators having been invented.

The security information presented to users by the user interface of communication channels such as emails, phone calls, etc., is even more important than those by web browsers in terms of helping users construct an accurate perception, because those are the communication channels where users' initial perceptions of the interaction are created. At this stage users have much less information selection bias and are more likely to pay attention to the information presented to them by the security tools/indicators. It is the best place to help users to discover the mismatch of participants, because the system designers could take full advantage of people's ability to construct an accurate perception.

Reduce the Burden On Users By Interpreting As Much Metadata As Possible

In the interaction model, after selecting the information users need to interpret the information. However, users may lack the knowledge to interpret messages presented to them at the user interface. This should be viewed as a design fault rather than the user's problem. In such cases, designers should ask the questions:

- What semantic information do users need to get?
- Can the system interpret the message for users so that the semantic information is presented to them in the most intuitive way?

For example, users are found to lack the knowledge to interpret URLs to understand who they are really interacting with [24]. A URL is a string of characters used to represent the location of an entity on the Internet. In server to user authentication, users are expected to confirm that they are indeed visiting the web site that they intended to visit. They need to parse the URL string in the URL address bar to obtain the identifier of the web site (i.e. the domain). A URL has five major components and ten sub–components, and different symbols are used to separate those components. The syntax and structure of the URL is illustrated in Figure 3.4.

Users without URL syntax education are likely to make mistakes in extracting identification information from a given URL, especially when the URL string is deliberately formed to deceive them, as happens in phishing and many other social engineering attacks on the Internet. Attackers

Figure 3.4: The Syntax of a URL

manipulate the URL links to create URLs that look like the legitimate counterpart, so that even when users do check the URLs they may still fall victim. To fill this knowledge gap in a relatively short period of time by user education is infeasible, because the population of users seeking access to the web is becoming increasingly large and diverse. It is not only expensive to educate them, but also difficult to reach them and get their attention. However, this problem could be relatively easy to solve by improving the design of how a URL string is displayed to user in URL the address bar.

First, let us analyse the components of a URL and what they are used for. A user's task is to extract the identification information (often identifier of an entity) from the given URL. To whom a resource belongs is identified by the domain, and optionally proved by a digital certificate. Among the five components of the URL (scheme, authority, path, query and fragment), only 'authority' contains the domain information. The other four components are all parameters concerning the implementation details of how the entity should be accessed.

'Scheme' specifies the communication protocol to be used, such as 'Ftp', 'Http' or 'Https', etc. 'Userinfo' is used to decide whether the current user can access the requested resource. 'Port', 'path', 'query', and 'fragment' are all concerned with the implementation details of the remote server. In addition, not all the strings contained in the hostname are relevant, sub–domains can also be considered as implementation details of the remote server; only the top level domain of the hostname should be used as the identifier. It is clear that the data contained in the other four components and the sub–domains are all data that are intended to be read and processed between the web browser on the client side and the server rather than by general users. If the users do not need them, can

the URL address bar just hide them from users? Two important software engineering principles are encapsulation and data hiding, as they provide much clearer readability and usability. Can the same principles be applied here?

When the web browser was invented, most of its users were technical users such as research scientists. The navigation of the Internet was mainly based on the manual input of URLs into the address bar. To do so, users often had to remember the URLs of important entities, and those users were expected to have sufficient knowledge to understand the syntax of URLs. As a result the URL is displayed as it is without any processing. However, the user groups have changed dramatically; now technical users are a minority, and the most common navigation method is to use a search engine (in this case there is no need to read the URL of a resource and remember it). Even when users do enter URLs directly, they are normally very short. As a result the main use of a URL has shifted to help users identify the identity of a resource. Would it not therefore be better if the web browser does the interpretation work for the users and hides the irrelevant implementation details from users as default, and lets technical users choose if they still want the URL to displayed as it is? Users can still input the URL in the same way, the difference is that after the URL has been typed in and processed, the URL address bar then hides unnecessary information. General users would no longer require knowledge to interpret the URL syntax and there would be much less room for attackers to play their link manipulation tricks.

3.5.2 Evaluation of User Interfaces

Current evaluations [71, 77, 88] of user interfaces have mainly focused on whether the users can select sufficient information provided to them at the interface and how well they can interpret the information presented to them. From our model we can see why those two aspects are very important, but two aspects seem to have received insufficient attention by researchers:

- whether the information provided at the interface can be tampered with by attackers. If so, then it is also important to check whether users can get alternative information sources to verify the information displayed at the user interface;

- whether the user interface provides enough metadata for users to accurately define the participants of the interaction.

The first addresses whether users will have correct information to make decisions. If the information used in the decision is wrong, how can users form accurate perceptions and discover mismatches? The second examines whether users have enough information to choose from. As discussed in the model, it is not that straightforward to decide what is sufficient information. Unreliable information displayed at the user interface needs more information to verify it and this increases the total information needed. If users do not have enough information to consider, forming accurate perceptions is impossible and exploitable mismatches will not be detected.

3.5.3 User Education

Most existing phishing user education research [73, 17, 18, 25, 30, 57, 62] is focused on helping users detect phishing attacks. Users are generally taught what existing phishing attacks look like via real and artificial examples, and the most common spoofing techniques used. Users are given guidelines to follow, such as: never click on URL links embedded in emails; do not trust URLs containing IP addresses, etc. However, phishing attacks are constantly evolving, they may use novel tricks or may be launched over communication media the user does not associate with phishing. Such education may improve the chances for users to detect the known phishing attacks, but it fails for the more sophisticated or unknown attacks. With more and more techniques available to attackers, the guidelines and blacklists would eventually become too complicated to follow. If users' education is too close to the current phishing attack techniques, and the techniques change then users may be more vulnerable than before.

Rather than injecting a new step, detection, into a user's behaviour model, we should focus on educating users to form an accurate perception when interacting with computing systems by:

- teaching them the sufficient set of information they should select in different context;
- teaching them how to correctly interpret the information selected.
- training users to have more complete/accurate expectations.

In this way users will have a better chance of discovering the mismatches we discussed earlier. By teaching users how to more accurately form perceptions, they can now protect themselves against both known and unknown attacks. This approach is also simple and independent from changing attacking techniques, and most importantly this approach is more likely to have consistent results as it does not change the fundamental model of how users interact with computing systems.

My model also suggests there are distinct limits to what can be achieved by education. User education improves only how accurately users form perceptions with the information available. The sufficiency and reliability of the information also affect the perceptions users construct, and those two aspects can only be addressed from a technical point of view. If reliable and sufficient information is not available, users cannot construct accurate perceptions and may still fall victim to phishing attacks. But it is wrong to blame users for security compromises and to tar them as "the weakest link" of security. As designers of security products and systems, we have been accessories to the crime!

3.6 Discussion

In this chapter I have introduced a user-phishing interaction model from a decision making point of view. This model is used to analyse how users can detect phishing attacks and discuss why users often fail to do so.

As mentioned in the previous chapter, there is also a model of phishing deception detection proposed by Wright et al [87] based on the model of

deception detection by Grazioli [37]. Wright's model focuses on how a deception could be discovered from the user's point of view. It focused on the user side and ignored the influence of the system interface on the outcome of the communication. As a result, the potential countermeasures followed by his model would be improve users' web experience and educate users to be more suspicious. These suggestions are well motivated, but were known before the introduction of this model. This model also has limited use for system designers in terms of how to design secure but also usable interfaces.

Another important limitation of Wright's model is that it does not consider the fact that user-phishing interaction comprises a *series* of actions; there may be several decisions users need to make and users' ability to detect deception may not be the same at different stages of the interaction. (Wright's paper did, however, state that the participants are more likely to detect deception during interaction with phishing emails than with phishing web pages.)

In contrast, the model described in this chapter captures the whole user-phishing interaction in an objective manner with equal emphasis on both users and information available at the user interface. It is based on the analysis of user behaviours in reported phishing attacks and established decision making theory. It could be a powerful, reliable predicative tool for security practitioners and system designers.

Chapter 4

Threat Modelling

This chapter introduces methods to identify, and assess user-related vulnerabilities within internet based user authentication systems. The methods described in this chapter can be used as an addition to risk assessment processes such as the one described by ISO 27001 [5].

4.1 Introduction

Threat modelling is a tool to identify, minimize and mitigate security threats at the system design stage. However, most existing threat modelling methods appear to give little in the way of systematic *analysis* concerning the user's behaviours. If users are now the "weakest link" then user-side threat modelling is as important as system-side threat modelling. In this chapter, a threat modelling technique for internet based user authentication systems is described. To improve practicability, the technique is designed to be quantifiable, and can be easily integrated

into one of the most prominent security and risk management methods described in ISO/IEC 27001 [5].

4.2 Overview Of The Method

The method comprises the four steps listed below. The method presented here has similarities with the overall threat modelling approaches of ISO27001 [5], but seeks to provide user centred threat interpretation.

1. **Asset identification**: identify all the valuable assets (authentication credentials which can be used to prove a user's identity) that attackers might obtain from users.

2. **Threat Identification**: identify threats the authentication credentials (assets) face based on the properties of these authentication credentials.

3. **Vulnerability and risk level assessment**: for each threat identified apply the corresponding vulnerability analysis technique to reveal the potential vulnerabilities and associated risk levels.

4. **Threat mitigation**: determine the countermeasures for each vulnerability identified. (This step is not within the scope of my research and how to determine countermeasures will not be discussed here.)

The terms threats and attacks are over-used. To help readers better understand the method, it is necessary to explain what these terms mean

here. I define threat as the possibility of an event that has some negative impact to the security of the system happening. An attack is a means through which a threat is realised. An attack is possible only if certain vulnerabilities exist in the system.

The threat modelling method described in this chapter takes an assets-centric (authentication credentials in this case) approach rather than the system-centric or attack-centric approaches. Authentication credentials are simple to identify, small in quantity, and can be an ideal focal point to conduct threat modelling. The foundation for system–centric approaches is knowledge of the implementation of the system and how it operates. Such an approach is not suitable when users are involved, because we do not possess sufficient knowledge to accurately predict users behaviours in all possible conditions. The attack–centric approach could be too complex given there are so many existing types of attacks and new ones emerge all the time as attackers are constantly adapting their techniques to increase success rates and avoid detection.

The assets-centric approach itself is not the author's original contribution. Rather, the contribution is a demonstration of how this approach can be applied to analyse the user related threats. For instance, how to identify and classify the assets, how to use the properties of the assets to identify vulnerabilities etc. The method described in this chapter should be also viewed as a supplement to the standard risk assessment processes described by ISO 27001. With the addition of this method, security professionals can still use the process they are familiar with to address the user-related vulnerabilities.

4.3 Asset Identification

All authentication credentials should be identified and grouped into sets, where each set can be used independently to prove a user's identity. If some credentials together can prove a user's identify to a server, then they should be grouped into one set. In general there are more than one set of user authentication credentials for each user account. The vulnerabilities and their associated risk levels identified by our method are specific to each authentication credential set. If any authentication credentials have been missed in this step, so are the associated threats and vulnerabilities.

It is simple to identify the authentication credentials such as user name and password that are used in the normal interaction between users and systems. (We call this set of authentication credentials primary authentication credentials.) Authentication credentials that are used in special cases are likely to be missed. Special cases include authentication during recovery/resetting of primary authentication credentials, and also authentication on communication channels other than the Internet. Some authentication credentials are also implicit, for example access to a secondary email account. It is very common that when someone forgets the primary password, he/she can get a new password sent to a chosen email account after answering a series of security questions correctly. In these cases, the access to the secondary email account should be also considered as a means of proving the user's identity. As a result, together with the security questions, the secondary email account should also be identified. In this method, the user identifier should also be treated as an authentication credential.

For each identified set of credentials, we recommend they be represented in the following way: { Identifier :: C_1, C_2, .. C_n } The identifier uniquely identifies a user in the given authentication system (the same id may identify a different user in another system), C_i represents a single authentication credential data item, such as a PIN or a password. The identifier may not always exist for the set whose members are not primary authentication credentials. Examples can be found in the case studies described in section 4.6 and section 4.7.

4.4 Threat Identification

Threat identification is based on the properties of the authentication credentials, which we will introduce first. Then we describe how to use those properties to discover the threats each set of authentication credentials faces.

4.4.1 Properties Of Users' Authentication Credentials

The approach requires that several properties of authentication credentials be identified and considered. The particular property value of authentication credentials affects which threats their use may be subject to. Five properties are given below. These properties affect, for example, which threats are relevant, how/who generates authentication credentials, as well as how they are shared and communicated.

- Factor;

- Assignment;
- Directness;
- Memorability;
- Communication channel;

Factor something users know (KNO) ; something users possess (POS); something users have access to (ACC); or characteristics of who users are (ARE).

'Something users have access to' is usually classified as 'something users possess'. It is distinguished because it has different implications for the security of an authentication system: it creates a security dependency relationship. Such authentication credentials are not directly possessed by a user, but the user holds the keys (authentication credentials) to access them. These credentials are managed and possessed by third parties, and their compromise can lead to the compromise of the concerned authentication credentials. For example, access to a secondary email account is often used as an authentication credential, it should be treated as ACC rather than POS. The email accounts are possessed and managed by email account providers and users only have access to them. A security breach on the email account can lead to the compromise of the studied authentication system, and the security of the email account is not within the control of the studied authentication system.

'Something users know' refers to a piece of information that users

remember (e.g. a password), while 'something users possess' refers to a physical object that users hold (e.g. a USB key, a SIM card). 'Who users are' refers to properties which can be used to describe the identity of a user, typical examples are: biometric data such as finger print, facial appearance of the user, etc.

Assignment by the server; by the user; or by a third party.

Assignment by the system can ensure that certain security requirements are met (for example, that values of the authentication credentials are unique, and difficult to guess). A user may not find the value assigned by the system usable. User defined values may have high usability, but the system has limited control over whether necessary security requirements are met. When the value is assigned by a third party, the security properties depend on the behaviour of the third party. If the value of the authentication credential is predictable or easy to replicate, then this vulnerability could lead to the compromise of the current system.

Directness direct or indirect.

This indicates what messages are exchanged between the user and the server: 'direct' means the authentication credential itself (including the encryption of the authentication credential) is exchanged; 'indirect' means a special message, which only the authentication credential is capable of producing, is exchanged, e.g., a response in a challenge which only can be solved by the holder of a secret.

Memorability high, medium or low.

Table 4.1: Property Relevance Table

Factor	Assignment	Directness	Memorability	CC
KNO	Yes	Yes	Yes	Yes
POS	Yes	Yes	No	Yes
ACC	Yes	No	No	No
ARE	No	Yes	No	Yes

This indicates how easy the value of an authentication credential is to remember.

Communication Channels (CC) open or closed.

This indicates whether the media or communication channels over which credentials are to be exchanged is implemented using agents that are not controlled and protected by the authentication server. For example the Internet is an open CC, because the user's machine, local routers, DNS servers and etc. are not controlled and protected by any authentication server.

The property identification process starts by identifying the factor on which the credential is based, then the factor property decides what properties to consider. Table 4.1 shows what properties should be considered for each factor. Finally for authentication credentials whose directness is 'direct', then the communication channel property needs to be considered.

4.4.2 Threat Identification Predicates

To determine the threats a credential faces one can apply its properties to a series of logical predicates we define. Currently there are three threat predicates identified :

server impersonated (SIT) This represents the threat that attackers impersonate a trustworthy server entity that has a genuine need for the authentication credentials, to elicit the authentication credentials from users. Phishing and Pharming attacks are typical examples of attacks that can realise this threat.

disclosure by users (DUT) This represents the threat that the authentication credentials are unwittingly disclosed by the users, where no deception is involved.

disclosure by a third party (DTPT) This represents the threat of disclosure of a user's authentication credentials because of security breaches of a third party.

Before we present the predicates we developed, we describe the basic syntax used to express them:

X :: the current authentication credential being considered

$X \cdot Property$ :: the value of the corresponding property

$X \in SIT/DUT/DTPT$ :: means X faces the given threat

$X \Rightarrow Y$ When X is true then Y is true

The predicates to identify the potential threats are as following:

$$\begin{array}{ll} & (X \cdot CC \equiv open) \\ \wedge & (X \cdot Directness \equiv direct) \end{array} \Rightarrow X \in SIT$$

$$\begin{array}{ll} & ((X \cdot Factor \equiv KNO) \wedge (X \cdot Memorability \neq high)) \\ \vee & (X \cdot Factor \equiv ARE) \\ \vee & (X \cdot Assignment \equiv user) \end{array} \Rightarrow X \in DUT$$

$$\begin{array}{ll} & (X \cdot Factor \equiv ACC) \\ \vee & (X \cdot Assignment \equiv thirdparty) \end{array} \Rightarrow X \in DTPT$$

Using the predicates and the properties of authentication credential we can determine the potential threats to it. Each member of a given authentication credential set may face different threats. The threats that all members face take the highest priority in the next two steps, because a single attack realising the common threat could compromise the whole set. If an authentication credential set does not have a common threat, then it would take a combination of attacks realising various threats to compromise the whole lot. This also suggests an important authentication system design principle: members of a set of authentication credentials should be chosen with the consideration that they do not face a common threat; if a common threat can not be avoided, then designers should at least try to minimize the common threats that the set of authentication credentials faces. Following this principle could significantly increase the difficulty of obtaining the whole set of user authentication credentials.

The threats are identified by considering all attack examples collected. Although attempts have been made to be as comprehensive as possible, there might be threats that are not yet addressed by our predicates. To identify those missing threats one can easily create the predicate that is needed on the nature of the threat. In this Chapter our focus is about the methodology, not about the completeness of the threats identification predicates. The more the method is used, the more we expect to identify new predicates.

4.5 Vulnerabilities and Risk Level Analysis

There are two methods to determine the vulnerabilities and the risk levels for each identified threat, the choice of the method depends on whether the threat can be realised during a user-system interaction or not. In the former case (we will refer it as the user-system interaction case) we analyse the relevant user interface and system features, and in the other case (we will refer it as the security policy case) we examine the relevant security policies and security assumptions placed onto users. For the three threats described in the previous section, only SIT is user-system interaction case and the other two are security policy cases.

4.5.1 User-system Interaction Case

To discover the vulnerabilities that could be exploited to impersonate authentication servers, one has to identify all the use cases where authentication credentials are exchanged. We provide a method to help analysts

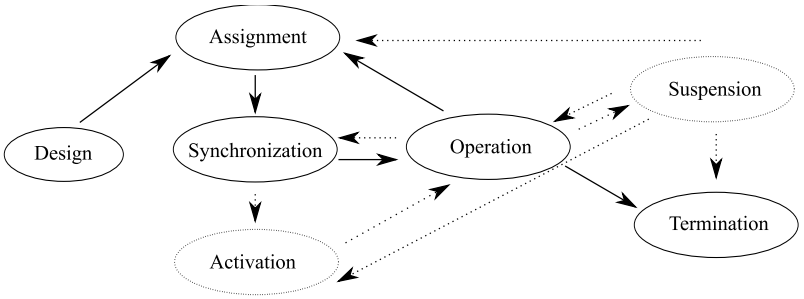


Figure 4.1: The Life Cycle of Authentication Credentials

consistently identify them with the help of an *authentication credential life-cycle*. Then for each use case, analysts can apply what is termed here the "user behaviour analysis" technique to uncover the potential vulnerabilities and associated risk levels.

The Life Cycle Of Authentication Credentials

Figure 4.1 shows states in the life cycle of authentication credentials and the transitions between them. The oval represents states. The optional state and transitions between states are represented by dashed lines. Following is the description of the states:

Design: Designers decide three things in this state : over which communication channel the authentication will be carried out; what authentication credentials should be used; and their life cycle. The design should consider the requirements for security, usability, economic constraints and the properties of the authentication credentials (described in Section 4.4.1). Threat modelling should also be carried out in this state.

Assignment: This state determines the value(s) of the authentication credential(s) for a particular user. Once this state completes, only the party who conducted the assignment knows the value of the credential(s). For example, in an authentication system where password, one time password generated by a USB key, and user name are used as the authentication credentials, these actions will be implemented by both users and servers. Users carry out the action to decide what the user name and password will be, while servers run a program to create a secret, and then store the secret in the USB key. In some cases this action can be completed almost instantly, for example if biometrics such as finger prints, facial appearance or voice are chosen.

Synchronisation: This state starts when user authentication credentials are assigned and ends when both user and server are capable of executing the user authentication process. During this stage, the party who assigned the value informs the other party of the value it has chosen. The time taken varies with the communication channel used. If users supply credential values via web pages then synchronisation could be almost immediate and the authentication credentials will be transitioned to other stages. For credentials exchanged by the postal system (e.g. a PIN for a new cash point card, or a USB token), then the synchronisation could take a few days. Sometimes when biometric information is used, users may need to travel to a certain location to securely give such information to the server.

Activation: Some systems may require users to take action to activate authentication credentials before they can be used. For example, it is not unusual to have to telephone a number to activate a banking account.

Operation: Users supply their primary authentication credentials to

authenticate themselves to external entities.

Suspension: The current authentication credentials temporarily cease to function, e.g. 'lockout' after three failed authentication attempts. Upon satisfying certain requirements authentication credentials can be transitioned to other stages.

Termination: Here current authentication credentials permanently cease to function. The account may have been terminated by the system or the user.

A given authentication system may not have all the states and actions mentioned. The activation action and suspension state are optional. Any authentication credential should pass through the remaining actions and states. Transitions between states may vary greatly for different authentication credentials. A typical authentication credential's life cycle starts at the design state before moving to assignment and synchronisation. Depending on the actual authentication system, there might be an activation action before the operation state. From operation it can reach: suspension, termination, assignment, and synchronisation. It often reaches assignment because the user or system decides to reset the current value of the authentication credentials. Not every system allows authentication credentials to transition from operation to synchronisation. But when it does, it is often due to loss of authentication credentials. For example, when a user forgets his password, the user asks the system to send him/her the password or a link that enables such a password to be established.

Using the life cycle, the following six situations have been identified where a user's authentication credentials could be exchanged: 1) Synchronisation State; 2) Operation State; 3) state transition from Operation to Assignment;

4) state transition from Operation to Synchronisation; 5) state transition from suspension to assignment; 6) state transition from suspension to operation. In practice an analyst does not need to consider the life-cycle of the authentication credential, they just need only to consider the six cases to find all use cases.

Vulnerability Analysis

The "user behaviour analysis" technique is based on the user-phishing interaction model described in chapter 3. In this model the interaction between user and system can be decomposed into a cycle of user's actions, and decisions. The interaction ends either when the interaction reaches its natural conclusion or when users discover they are being attacked. For each use case, analysts should use such an interaction model to describe the user-system interaction and decompose each use case into a series of security related user actions and decisions. For each action and decision the analyst then considers whether the vulnerabilities described in Table 4.2 exist. The vulnerabilities in the table are derived using the vulnerabilities proposed in [23, 53]. The judgement of the vulnerabilities caused by non-technical factors can be subjective, however, the analyst should always think from a worst case perspective and make decisions accordingly.

4.5.2 Security Policy Case

Analysts should first find out the security policy which has been put in place to deal with the concerned threat as well as the assumptions re-

Table 4.2: Vulnerability Table for User Action and User Decision (adapted from [23, 53])

	Apply To	Description
USV–A1	User action	Users are unable to understand which security security actions are required of them.
USV–A2	User action	Important security actions required are not enforced by the system design.
USV–A3	User action	Users do not have the knowledge or skills to retrieve sufficient information to make the right security decision.
USV–A4	User action	The mental and physical load of a security action is not sustainable.
USV–A5	User action	The mental and physical load of carrying out repeated security actions for any practical number of instances is not sustainable.
USV–D1	User decision	Users do not have sufficient knowledge to interpret the message presented to them at the user interface.
USV–D2	User decision	The system does not provide the user with sufficient information for making the right security decision.
USV–D3	User decision	The mental load of making the correct security decision is not sustainable.
USV–D4	User decision	The mental load of making the correct security decision for any practical number of instances is not sustainable.

Table 4.3: Vulnerability Table for Security Policy

USV–P1	Users are not aware of the security policy.
USV–P2	The system designers' assumption of what users would do is unrealistic or users can not obey the security policy due to the lack of knowledge, capabilities or other resources.
USV–P3	The security policy or assumption contradicts the current social context, common sense, and users' common behaviours
USV–P4	The security policy or assumption contradicts other systems' security policies.
USV–P5	The security policy is counter-productive for users' primary needs.
USV–P6	The assumption is wrong or users can not obey the security policy consistently over a long period of time because the mental and physical load of doing so is not sustainable.

garding how users would protect and use their authentication credentials. They should then check whether any of the vulnerabilities described in Table 4.3 exist in the studied system. The second step is subjective, but we recommend that analysts should take a worst case approach.

The following example can be used to explain USV–P3. Because of the popularity of social networking sites and web 2.0/3.0 sites, personal information such as name, address, and date of birth are becoming publicly displayed on the Internet. If the security policy requires users to keep such information private, then this security policy contradicts the current social context and users' common behaviours. As a result, this security policy is very likely to be breached.

USV–P4 holds if the security policy or assumptions differ from those of other systems. Any difference is likely to cause confusion and users will find it difficult to decide which policy should be obeyed in any particular situation. For example, to stop phishing attacks some financial organizations tell their customers that they will never send emails to them, while other organizations (especially Internet based banks) use email as a primary channel to inform offers or news to its customers. For a user of both types of organization, confusions may easily arise.

USV–P5 and USB–P6 are the result of lack of usability; they often lead users to cut corners or simply ignore the given security policy. They often exist at the same time. For example, the user authentication system for accessing the UK grid computing system requires each user to hold its own digital certificate and a strong password. Users are asked not to share these authentication credentials and to keep them secret. However, such certificates take a couple weeks to obtain, and applicants must bring their ID document to a local centre. Researchers often need to collaborate with international colleagues who cannot apply the certificates with ease. Some researchers may suddenly find that they need to use the grid computing resource to complete an important experiment. Many of them are reluctant to go through the lengthy application. To make their life easier many researchers share certificates and passwords. In some cases, a group of researchers may use the same authentication credentials. Given the diversity and mobility of the workforce, it is very easy to lose track of who has access to a certain authentication credential. It is easy for a malicious attacker to obtain those authentication credentials.

4.5.3 Risk Level Estimate

The risk level assessment is conventionally carried out by considering the difficulty of exploiting a vulnerability, the impact of the exploit, and the likelihood of the exploit. When the human aspect is considered, two extra factors also need to be put into equation:

Diversity within the targeted user group Each user has different knowledge, skills and behaviours. The knowledge of concern here relates to system implementation and security. Because of such differences, a vulnerability may apply to some users but not others.

Inconsistency and unpredictability of each individual user's behaviour A user's abilities and behaviours do not remain the same. Emotional state, physical conditions, current priorities, training and education may all affect a user's behaviours. Hence, a vulnerability may not always apply to the same user.

For each vulnerability I associate two probabilities P_u and P_i for these two factors. P_u represents the portion of the user group to whom this vulnerability applies, while P_i represents the probability that a vulnerable user falls victim to the attack on any one occasion. Since it is very difficult to measure the actual values for P_u and P_i, qualitative scale values: high (H), medium(M), and low(L) are used. P_u is high means the user group are very diverse in terms of ability and knowledge. There are five risk levels: Very High (5), High (4), Medium (3), Low (2), Very Low (1). Table 4.4 indicates how to determine the risk level. The first column of the table indicates how difficult it is to exploit the vulnerability and the second column('Yes' and 'NO') indicates whether the exploitation

Table 4.4: Risk Level Assessment Table

		$P_u \equiv$ H			$P_u \equiv$ M			$P_u \equiv$ L		
		$P_i \equiv$ H	$P_i \equiv$ M	$P_i \equiv$ L	$P_i \equiv$ H	$P_i \equiv$ M	$P_i \equiv$ L	$P_i \equiv$ H	$P_i \equiv$ M	$P_i \equiv$ L
Difficult	Yes	5	4	4	5	4	3	4	3	3
	No	2	2	2	2	2	1	2	1	1
Not Difficult	Yes	5	5	5	5	5	5	5	5	4
	No	3	3	2	3	2	2	2	2	1

of the vulnerability would lead to the compromise of the whole set of authentication credentials. For example, the top left cell means the risk level is very high (5) as the vulnerability is difficult to exploit but it can lead to the compromise of the whole set of authentication credentials and P_i and P_u are both high.

The values in Table 4.4 are chosen based on the security requirements for the user authentication system which will be discussed in case studies. In practice one could adjust the values in this table based on the security requirements of the system been analysed. This table is a framework for risk level assessment. This framework ensures the risk assessment process is consistent and comprehensive. It is also recommended that if the risk level is higher than medium, mitigation of such vulnerabilities should be considered. So when adjusting the values in this table, one should consider the above suggestion as well.

4.6 Case Study One

We illustrate the approach with reference to a case study based on the current online user authentication system used by one of the UK's largest retail banks. This demonstrates that the proposed method can be applied to discover serious threats and vulnerabilities in a consistent and systematic manner.

4.6.1 Assets Identification

There are three sets of user authentication credentials, and they are:

- Set A: { user id :: password , memorable word }
- Set B: { :: first name, last name, birth date, sort code, account number}
- Set C: { user id :: first name, last name, birth date }

Set A is used in primary authentication to allow access to the internet banking facilities, set B is used to retrieve the forgotten user id used in Set A and set C is used to reset the password and the memorable word used in set A. Set B is an example where a set of authentication credentials does not have an identifier, instead the all members of the set can be used to uniquely identify the account holder.

4.6.2 Threat Identification

Properties

Obtaining the properties for the authentication credentials in the three sets is simple and the results are shown in the Table 4.5, Table 4.6, and Table 4.7. The authentication credentials are exchanged via the Internet. We regard it as an unsecured communication channel despite the encryption used, because the platform users used to browse the internet could be compromised as could the security of the middle layer agents such as local router, DNS server etc.

Identifying the value for some properties can be subjective, for example, memorabilities. In those cases, practitioners should assume a worst case scenario.

Threats

Applying the properties of those authentication credentials we can identify the threats each authentication credentials face. The results are shown in Table 4.8, Table 4.9, and Table 4.10

In conclusion, all members in the three sets of authentication credentials face SIT, and DUT. The attacks realizing either threat can compromise the whole set of authentication credentials and allow attackers to impersonate the user. The vulnerabilities and risk levels associated with them must be analysed. The results (shown in Table 4.8, Table 4.9, and Table 4.10)

highlight the vulnerabilities in the authentication system design. In particular, members of the authentication credential set share at least two threats. As a result, a single attack exploiting either threat can compromise the whole set of authentication credentials. This constitutes a "single point of failure". In many areas of dependability, "designing out" single point of failure is the norm. Here this approach has merit. Removing or reducing the common threats will generally reduce risk. Secure design should try to make sure that there are no threats (or at least very few) common to the set of all member authentication credentials. Fewer common threats means that fewer attacks could compromise all members at once and increases the difficulty for attackers to obtain the information they want.

4.6.3 Vulnerabilities And Risk Levels

Only the SIT requires user-system interaction, and the DUT is dealt with by security policy.

User Behaviour Analysis

As described earlier, this analysis technique starts by identifying the use cases where authentication credentials are exchanged, and then discovers what user security actions and decisions need to be made in those use cases. Finally we analyse whether vulnerabilities exist in those actions and decisions.

First let's identify the use cases by reference to the six possible cases

described in section 4.5.1. For set A, the authentication credentials are exchanged via user-system interaction for the following cases: synchronization (during reset) and operation states. In the two use cases users are supposed to take security actions to decide whether they are interacting with the legitimate website. For sets B and C, the authentication credentials are exchanged only during the operation state. In all identified cases the authentication credentials are exchanged via the Internet at the bank's website, and the security mechanisms used for the server to user authentication system are the same. As a result the security user action and decision models are also the same in all cases. Hence, we need only analyse one case to identify all the vulnerabilities and risk levels.

The bank's website uses the Extended Validated Certificate (EVC) as the means to provide reliable server to user authentication. The required users actions are: read the certificate image displayed in the URL address bar (action 1); and read the URL string displayed in the URL address bar (action 2). The user decisions to be made are:

- determine whether the current website belongs to the bank by using the information displayed by the certificate image (decision 1);

- determine whether the current website belongs to the bank by interpreting the URL string (decision 2).

Since there is no mechanism to make sure action 1 is taken by users before user authentication can begin, it is obvious that it has vulnerability USV-A2. This vulnerability is not difficult to exploit and it could lead to the compromise of the whole set of authentication credentials. Since the users of the online banking are the general public and it is certain that

not many users possess appropriate technical skills and knowledge, both the P_u and P_i at least should be medium. By referring to the Table 4.4, we can decide that the risk level is at least high (4).

The EVC is the replacement for the conventional TSL certificate which has been proved ineffective in providing usable server to user authentication. The EVC requires users to read the text displayed and the colour of the certificate image. Although it might be much more usable and more resilient against spoofing attacks, it requires users to understand the difference between a conventional TSL certificate and the EVC to achieve the desired result. However, it would be unrealistic to assume that the majority of users hold the required knowledge. Users without such knowledge may not understand what the colour of the digital certificate means and what the lack of colour on it means. Users have proven poor at noticing missing components in the user interface [26, 77]. As a result the action 2 also has USV-A1 and USV-A3. Both vulnerabilities are easy to exploit and could lead to the compromise of the whole set of authentication credentials. The P_u for both vulnerabilities is high, as most users still do not possess the required knowledge, while P_i should be at least medium. So the risk level for both USV-A1 and USV-A3 in action 2 is very High (5). Although both vulnerabilities and risk levels are not static, the risk level would drop if more users had the required knowledge and the vulnerability could disappear if majority of users were aware of EVC. For the same reason mentioned here, decision 1 has USV-D1 and its risk level is very high (5).

As with action 1, action 2 also has USV-A2 with the same risk level. Many user studies have shown that users are aware that they need to check the URL of the website during the server to user authentication [24, 48], it does not have USV-A1 and USV-A3. Decision 2 has USV-D1,

because only users with knowledge of how URL syntax works know how to interpret a given URL to find out the top level domain name. Using the top level domain name, a user can then decide whether the current website is legitimate of not. Many users do not have such knowledge. The above analysis also suggests that the P_u and P_i are both high, so the risk level for USV-D1 is very high (5). Decision 2 also has USV-D3 and USV-D4. Many URLs of the legitimate web pages are complex and long. Even worse for users is that the phishing websites would deliberately make their URLs more complex and longer in order to cheat the victims. Occasionally careful examination of a URL to determine with whom the user is interacting will be tolerable, but frequent detailed examinations will not. Alos, most examination efforts are not necessary (most websites users interact with are legitimate). USV-D3 and USV-D4 are also very high (5).

Security Policy Vulnerability Analysis

As threat identification has revealed that the bank's authentication system faces DUT. The bank has told its users that they should keep the authentication secret and not share them with anyone.

The authentication credential set A requires a user to remember an online banking ID, a password and a memorable word. The ID is a 8 digit number which holds no meaning for users and can be difficult to remember. The password is any combination of 6 - 15 letters and/or numbers, while the memorable word must contain between 6 and 15 characters with no spaces and contain both letters and numbers. The three pieces of information together might be difficult to remember for some users and

as a result they could write them down or store them externally. Once they have done that they risk the information being exposed to attackers; as a result we consider that it has USV–P2. This vulnerability is difficult for attackers to exploit systematically, although it can still lead to the compromise of the whole set of authentication credentials. The P_u is low and P_i is high, so its risk level is medium (3).

The sets B and C do not have USV–P2, but they do have USV–P3 and this is much more serious. The first name, last name, and date of birth are no longer secret. Such information may openly be displayed on social network web pages, personal blogs, etc. Users are not reluctant to give away such information and indeed many web sites ask users for them. Such personal information is also no secret to a user's friends and relatives. The account number and sort code is not secret either. People often given those information to others when they want to exchange money from their bank account, e.g. paying rent, share bill, loan, etc. The difficulty of exploiting this vulnerability depends on whether the attacker has a relationship with the victim or not, if they have then it is very easy otherwise could be difficult. Both P_u and P_i are high, so the risk level can be very high (5) for insiders(attackers who have relationships with victims) or high (4) for normal attackers.

It is quite surprising that the authentication system has such a serious vulnerability. As a result, insiders, such as the user's friend, colleagues or family member can easily get hold of the required authentication credentials to access the user's online banking account.

4.7 Case Study Two

In the second case study we analyse the user authentication system of Ebay. Ebay's customers are frequently targeted by phishing attackers, as a result one would assume their authentication system should be robust by now.

4.7.1 Assets Identification

For each user there are three sets of authentication credentials, and they are:

- Set A: { user id :: password}
- Set B: { email address :: access to the email address }
- Set C: { user id:: answers to a preset security question, post code, telephone number, birthdate }

Set A contains the primary authentication credentials. Set B is used to retrieve the user id used in the set A, and set C is used to reset the password used in set A. The compromise of either set B and C can not give attackers access to the account. There are six different security questions users could choose in set C and they are:

- What street did you grow up on?

- What is your mother's maiden name?
- What is your maternal grandmother's name?
- What is your first girlfriend/boyfriend's last name?
- What is the name of your first school?
- What is your pet's name?

4.7.2 Threat Identification

Properties

The properties for the authentication credentials in the three sets are shown in the Table 4.11, Table 4.12, and Table 4.13.

Threats

Considering the properties of those authentication credentials we can identify the threats that each authentication credential faces. The results are shown in Table 4.14, Table 4.15, and Table 4.16

4.7.3 Vulnerabilities And Risk Levels

Authentication credentials in set A , Band C are all facing SIT, and Set A and C are also facing DUT. The access to the email account in set B also faces DTPT.

User Behaviour Analysis

With reference to the six possible cases described in section 4.5.1, we can identify the following use cases for the three sets of authentication credentials. First for set A, the authentication credentials are exchanged via user-system interaction for the following cases: synchronization (during reset) and operation states. In these two use cases users are supposed to decide whether they are interacting with the legitimate website. For sets B and C, the authentication credentials are also exchanged in the operation and synchronization states. In all identified cases the authentication credentials are exchanged via the Internet at the Ebay's website, and the security mechanisms used for the server to user authentication system are the same. As a result the security user action and decision models are also the same in all cases, hence, we need only to analyse one case to identify all the vulnerabilities and risk levels.

The Ebay's website uses the Extended Validated Certificate (EVC) as the means to provide reliable server to user authentication. The user's actions to be taken are: read the certificate image displayed in the URL address bar(action 1), and read the URL string displayed in the URL address bar (action 2). The user decisions to be made are: determine whether the current website belongs to Ebay by using the information displayed

by the certificate image (decision 1), and determine whether the current website belongs to Ebay by interpreting the URL string (decision 2).

It appears that both the server to user authentication system and the user action and decision model are the same as the ones in the case study one, so we will not analyse them again here. In fact, any EVC based user authentication shares the following vulnerabilities: USV-A2 and USV–D1. The associated risk level for USV-A2 is high (4) and for USV-D1 is very high (5). The server to user authentication by reading the URL string shares the following vulnerabilities: USV-A1, USV-A3, USV-D3, and USV-D4. The associated risk level for all four vulnerabilities is very high (5).

Security Policy Analysis

The authentication credentials in set A and C face DUT. Ebay has no particular security policies for authentication credentials apart from telling its users they should choose a strong password and keep the authentication credentials secret.

The authentication credential set A requires the user to remember a user name and a password. The user name is an alphanumeric string which is assigned by users. The password must be at least 6 characters long and is a combination of at least two of the following: upper-case or lower case letters (A-Z or a-z), numbers (0-9) and special characters. By checking Table 4.3, the choice of such authentication credentials could lead to USV-P2 and USV-P3. Both authentication credentials are assigned by users, hence reuse of the authentication credentials can not be prevented. Given

users have over 20 active online accounts (using a password as one of the user authentication credentials) on average, most users would not have the ability to remember distinct passwords for each account. It is very likely some of them will write passwords down, store them externally, or even reuse them. As a result the security policy requirement to keep the authentication credentials secret would be broken. However, the risk level associated with these two vulnerabilities are low (2), because first these vulnerabilities may not lead to the compromise of the whole set of authentication credentials, secondly it is still difficult to find out where users have reused the authentication credentials, and thirdly the value of P_u and P_i are medium at most.

The composition of set C is very similar to the set C in case study one, they are based on information describing who users are. As with set C in case study one, it also has vulnerabilities USV-P3 and USV-P4. The associated risk level is very high for insiders and high for normal (external) attackers.

The access to an email account in set B faces DTPT. Attacks realising only this threat could not compromise the whole set of authentication credentials in Set A. This type of threat hasn't yet been discussed. Since the email account is managed by a third party chosen by a user, the security of the email account itself and the authentication credentials to access that email account are uncertain to Ebay, and Ebay has no control over users' choice of email account providers or the security of the email account provided. The compromise of the email account can lead to the compromise of set B. As a result, the USV-P7 exists. P_u and P_i are difficult to determine, because of the uncertainty introduced by the user's choice. With common sense, most people use a reasonably protected email account provider such as google, hotmail, yahoo, etc, so P_u is low

(users who use less secure email providers are minority). However, if for users who do use less secure email providers, then they are likely to be attacked. So P_i should at least be medium. By checking Table 4.4, the risk level for USV-P7 is very low(1).

4.8 Discussion

In this chapter, a threat modelling method to identify user related vulnerabilities within an authentication system is described. This method is asset (authentication credentials) centric, and has four steps. In the system design stage, designers can follow the steps and methods described to analyse how attackers could make users to behave in ways that could lead to the compromise of authentication credentials. The method is also designed to make the execution as systematic as possible. For example, the various tables can provide a consistent framework and procedure for execution. The usefulness of this method is also demonstrated by the two case studies.

One of the difficulties of using this method is how to assign the appropriate value to the properties of authentication credentials, or P_u and P_i etc. This needs to be determined based on the characteristics of the user group. If such knowledge is not directly available, then one has to make assumptions. If in doubt then the analyst should assume worst case scenario. The experience of security professionals could also make this judgement easier, but this would only be possible if the method is applied to study more authentication systems.

Table 4.5: Authentication Credential Properties for Set A

		user id	password	memorable word
Factors	KNO	√	√	√
	POS			
	ACC			
	ARE			
Assignment	User		√	√
	System	√		
	Third Party			
Directness	direct	√	√	√
	indirect			
Memorability	High			
	Medium	√	√	√
	Low			
CC	Closed			
	Open	√	√	√

Table 4.6: Authentication Credential Properties for Set B

	Factors				Assignment			Directness		Memorability			CC	
	KNO	POS	ACC	ARE	User	System	Third Party	direct	indirect	High	Medium	Low	Closed	Open
first name				✓	-	-	-	✓		-	-	-		✓
last name				✓	-	-	-	✓		-	-	-		✓
birthdate				✓	-	-	-	✓		-	-	-		✓
sort code	✓					✓		✓				✓		✓
account number	✓					✓		✓				✓		✓

Table 4.7: Authentication Credential Properties for Set C

		user id	first name	last name	birthdate
Factors	KNO	✓			
	POS				
	ACC				
	ARE		✓	✓	✓
Assignment	User		–	–	–
	System	✓	–	–	–
	Third Party		–	–	–
Directness	direct	✓	✓	✓	✓
	indirect				
Memorability	High		–	–	–
	Medium	✓	–	–	–
	Low		–	–	–
CC	Closed				
	Open	✓	✓	✓	✓

Table 4.8: Threats for Set A

	SIT	DUT	DTPT
user id	√	√	
password	√	√	
memorable word	√	√	

Table 4.9: Threats for Set B

	SIT	DUT	DTPT
first name	√	√	
last name	√	√	
birthdate	√	√	
sort code	√	√	
account number	√	√	

Table 4.10: Threats for Set C

	SIT	DUT	DTPT
user id	√	√	
first name	√	√	
last name	√	√	
birthdate	√	√	

Table 4.11: Authentication Credential Properties for Set A

		user id	password
Factors	KNO	✓	✓
	POS		
	ACC		
	ARE		
Assignment	User		✓
	System	✓	
	Third Party		
Directness	direct	✓	✓
	indirect		
Memorability	High		
	Medium	✓	✓
	Low		
CC	Closed		
	Open	✓	✓

Table 4.12: Authentication Credential Properties for Set B

	Factors				Assignment			Directness		Memorability			CC	
	KNO	POS	ACC	ARE	User	System	Third Party	direct	indirect	High	Medium	Low	Closed	Open
email address	√				√			√		√				√
access to the email account			√				√	–	–	–	–	–		√

Table 4.13: Authentication Credential Properties for Set C

	Factors				Assignment			Directness		Memorability			CC	
	KNO	POS	ACC	ARE	User	System	Third Party	direct	indirect	High	Medium	Low	Closed	Open
user id	√					√		√			√			√
answers to security question				√	–	–	–	√		–	–	–		√
post code				√	–	–	–	√		–	–	–		√
telephone number				√	–	–	–	√		–	–	–		√
date of birth				√	–	–	–	√		–	–	–		√

Table 4.14: Threats for Set A

	SIT	DUT	DTPT
user id	√	√	
password	√	√	

Table 4.15: Threats for Set B

	SIT	DUT	DTPT
email address	√	√	
access to the email account	√		√

Table 4.16: Threats for Set C

	SIT	DUT	DTPT
user id	√	√	
answers to security question	√	√	
post code	√	√	
telephone number	√	√	
date of birth	√	√	

Chapter 5

User Behaviours Based Phishing Website Detection

This chapter presents the design, implementation and evaluation of the *user-behaviour* based phishing detection system (UBPD), a software package designed to help users avoid releasing their credentials inappropriately.

5.1 Overview

UBPD alerts users only when they are about to submit credential information to a phishing website (i.e. when other existing countermeasures have failed), and protects users as the last line of defence. Its detection algorithm is independent of the manifestation of phishing attacks, e.g., how phishing attacks are implemented and deception method used. Hence, its detection is resilient against evasion techniques, and it has significant

potential to detect sophisticated phishing websites that other techniques find hard to deal with. In contrast to existing detection techniques based only on the *incoming* data (from attackers to user victims), this technique is also much simpler, and needs to deal with much less low level technical detail.

Note: In this chapter we use 'interact', 'interaction' and 'user-webpage interaction' to refer to the user supplying data to a webpage.

5.2 Detection Principle

The work described in the previous chapter aims to discover the threats arising at authentication interfaces. Yet despite our best efforts to reduce the risks associated with interfaces, we have to accept that false perceptions will be created and render users exploitable. Thus we need to address these aspects too.

From a detection system's perspective, phishing attacks can be described by the simple model shown in Figure 5.1 (this model is abstracted from the attack incidents collected and phishing attack techniques the author is aware of). A phishing attack has two parts. In the first part attackers try to send user victims into the phishing attack black box by approaching them via a chosen communication channel. (The most frequently used communication channel is email.) A phishing attack is a black box because prior to the launch of the attack, victims and detection systems have no knowledge of the strategy of the attack, what techniques will be used, and how phishing email and websites are implemented. After going through

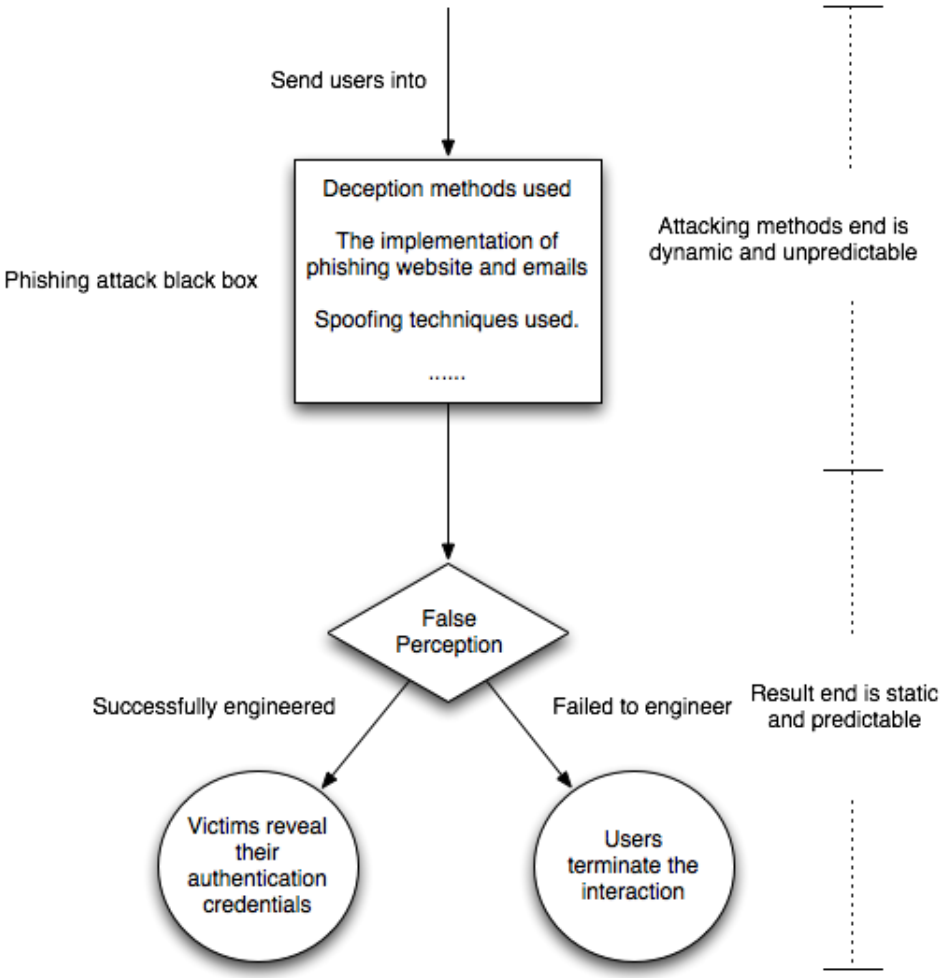


Figure 5.1: Existing Phishing Attack Detection Model

this black box, in the second part of this model there are two possible outcomes for a user: a false perception has successfully engineered and suggested actions taken, or a false perception has not been engineered and suggested actions have not been taken.

Existing detection methods discussed in Chapter 2 try to understand all possible internal mechanisms of this black box (how attackers deceive victims to disclose sensitive information, e.g, the implementation of phishing attacks, characteristics of phishing email and phishing websites, and deception methods used), and then in turn recognise manifestations of such attacks among many other legitimate user-system interactions. This approach to detection is reactive and high detection accuracy can be very difficult to achieve, because:

- It is very hard to lower the false positive rate while keeping the detection rate high. It may be very hard to distinguish phishing websites and legitimate websites by examining their contents from a technical point of view. A phishing website itself can be implemented in the same way as a legitimate website.

- Attackers have the advantage in evading existing detection by changing and inventing new mechanisms in the black box, i.e. varying the deception method as well as the implementation of the phishing websites.

To detect phishing websites effectively and efficiently, a fundamental change to the existing detection approach should be considered. According to the model described in chapter 3, the last step before a phishing attack succeeds is the execution of the actions (releasing of sensitive information) suggested by attackers. Hence UBPD aims to detect when a

user is about to release his/her credentials to attackers. It takes the view that the problem is really the release of credentials to inappropriate places; if there are no unfortunate releases. there is no problem. Furthermore, since the release of credentials is at the core of many phishing attacks, the approach should find wide application. Hence, the detection is independent of attacking techniques. The advantage is that an attacker's strategy can no longer affect the detection, and it could potentially achieve a very low false positive rate while keeping the detection rate high as well. Another advantage is that UBPD's detection is on demand, it will only actively interrupt users when the user is about to carry out counter-productive actions. This leads to fewer interruptions for a user and better usability as result of this.

5.2.1 What User Actions Should UBPD Detect?

Phishing websites are illegal and they work by impersonating legitimate websites. To reduce the chances of being discovered, attackers would only host phishing websites when an attack is launched. Phishing websites are also constantly being monitored, and once discovered they are taken down. As a result, phishing websites have a very short life time. On average a phishing website lasts 62 hours [66]. This suggests that users are extremely unlikely to visit a phishing website prior to the point of being attacked. So the first action UBPD examines is whether a user is visiting a new website or not. However, visiting a new website can not be used in isolation to decide whether a user is being attacked, and so UBPD further monitors users' actions when they are visiting a new website.

Secondly, regardless of methods used, as described in chapter 3 phishing

attacks always generate an erroneous user perception. In successful web based phishing attacks, victims have believed they are interacting with websites which belong to legitimate and reputable organisations or individuals. Thus the crucial mismatch that phishers create is one of real *versus* apparent identity.

In UBPD mismatch detection is informed by comparisons of current and previous behaviours. The authentication credentials, which phishers try to elicit, ought to be shared only between users and legitimate organisations. Such (authentication credential, legitimate website) pairs are viewed as the user's *binding relationships*. In legitimate web authentication interactions, the authentication credentials are sent to the website they have been bound to. In a phishing attack the mismatches cause the user to unintentionally break binding relationships by sending credentials to a phishing website. When a user is visiting a new website and also tries to submit the authentication credentials he or she has already shared with a legitimate website, UBPD decides that the user is currently visiting a phishing website.

So in summary, a phishing website can be detected when both of the following two conditions are met:

1. the current website has rarely or never been visited before by the user;

2. the sensitive data, which the user is about to submit, is bound to a website other than the current one.

The first condition is relatively simple to check, the difficulty lies in how

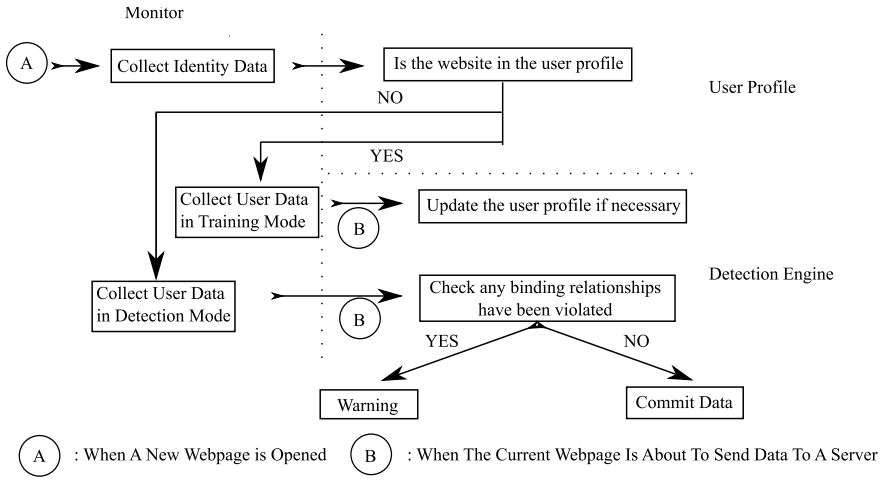


Figure 5.2: Detection Process Work Flow

to accurately predict whether the second condition will be met before any data is transmitted to a remote server. The following sections describe how UBPD is designed to achieve this.

5.3 System Design

This section describes the design of the UBPD and provides an overview of how information flows through the system, how user behaviour related information is created and maintained, how the information is used to determine the risks of the current interaction, how phishing sites are detected, and how the security of the system is maintained.

5.3.1 Overview Of The Detection Work Flow:

UBPD has three components:

- The user profile contains data to describe the user's binding relationships and the user's personal white list (this is a list of sites that the user has visited frequently). This profile is constructed once UBPD is installed, so that UBPD can detect phishing websites immediately.

- The monitor listens to actions by the user, collects the data the user intends to submit and the identity of the destination websites, and activates the detection engine.

- The detection engine uses the data provided by the monitor to detect phishing websites and update the user profile when necessary.

UBPD has two working modes: training mode and detection mode. In training mode, UBPD runs in the background, and focuses on learning newly created binding relationships or updating the existing binding relationships. When in detection mode, UBPD checks whether any of the user's binding relationships would be violated if the user-submitted data is sent to the current website. The mode in which UBPD runs is decided by checking whether the webpage belongs to a website

1. whose top level domain[1] is in the user's personal white list; or

[1]Suppose the URL of a webpage is "domain2.domain1.com/files/page1.htm", the top level domain is "domain1.com"

2. with which the user has shared authentication credentials.

There could be cases where a website is not in user's white list but with which the user has shared authentication credentials. Hence, UPBD checks both cases to decide the running mode. If either is true the system will operate in training mode, otherwise, it will operate in detection mode. Potential phishing webpages will always cause UBPD to run in the detection mode, since they satisfy neither condition. The legitimate websites with which users have binding relationships always cause UBPD to run in the training mode. UBPD will be running under detection mode if a user is visiting a new website.

The detection work flow is shown in Figure 5.2. Once a user opens a new webpage, the monitor decides in which mode UBPD should run. Then, according to the working mode the monitor chooses an appropriate method to collect the data the user has submitted to the current webpage, and sends it to the detection engine once the user initiates data submission. The details of the data collection methods are discussed in section 5.4. When running in detection mode if the binding relationships are found to be violated, the data the user submitted will not be sent and a warning dialogue will be presented. For the remaining cases, UBPD will allow data submission.

5.3.2 Creation Of The User Profile:

The user profile contains a personal white list and the user's binding relationships. The personal white list contains a list of top level domains of websites. The binding relationships are represented as a collection of

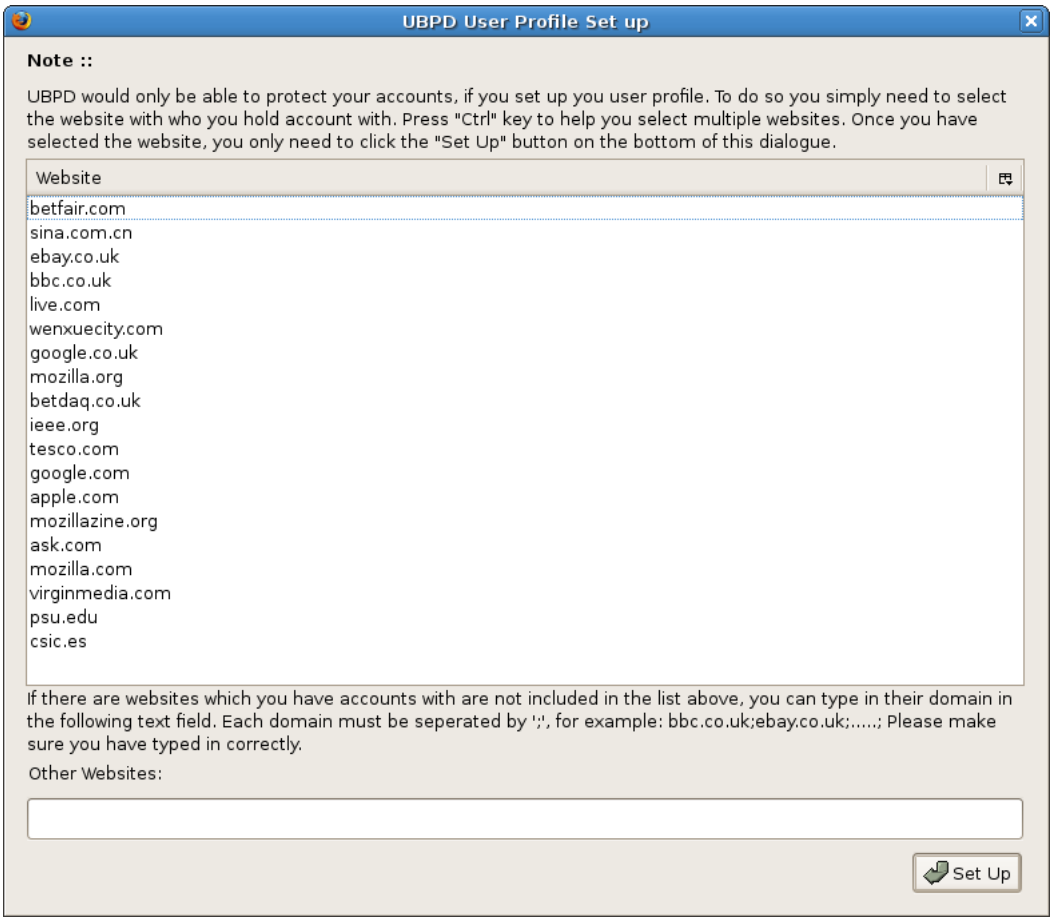


Figure 5.3: User Profile Creation –1

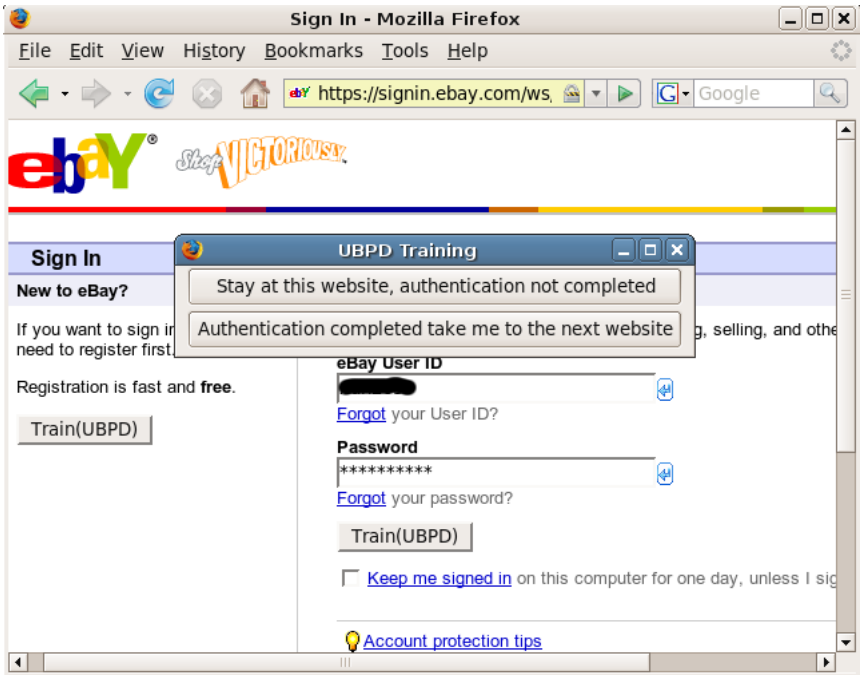


Figure 5.4: User Profile Creation –2

paired records, i.e., $\langle aTopLevelDomain, aSecretDataItem \rangle$.

Creation of a user profile is a mandatory step of UBPD's installation procedure. This is security action users must take to ensure the detection accuracy. Suggested by the threat modelling method described in chapter 4, if user security action is not enforced by the design, a vulnerability will be created.

Having successfully installed, UBPD will compose a personal white list when the web browser is restarted for the first time. This white list is mainly used to monitor an occurrence of the first user action mentioned in section 5.2.1. It is worth emphasizing that this white list is not used to decide whether the current website is legitimate or not. The top level domain names of the websites, which the user has visited more than three times (configurable) according to the user's browsing history, will be added to the white list. In addition the top level domain names of the most visited 500 (configurable) websites in the user's country are also added. This information is obtained from Alexa, a company specialising in providing internet traffic ranking information. By default, this white list is also automatically updated weekly.

The addition of the most visited 500 websites improves system operating efficiency (reducing the number of times the system needs to run in detection mode and the number of occasions where users need to interact with the warning dialog). As already discussed in Chapter 4, if a security action happens frequently, then this could lead to unacceptable mental load on users. As a result users may users pay less attention to it and could make wrong decisions. So it is very important to reduce the number of times users receive the warning message. These 500 websites could be viewed as a reflection of mass users' online browsing behaviours, and

they are clearly not phishing websites. Given these websites' popularity, if a user did not visit and have an account with them already, the user is likely to do so in future. Without this addition these websites will make UBPD run under detection mode when a user is visiting these websites for the first time. With this addition, UBPD would run in training mode in these circumstances. The computation required in training mode is much less intense than in detection mode. Moreover, in detection mode if a user reuses existing authentication credentials to set up an account with these websites during the user's first visit, UBPD will issue a phishing website warning. Although such reuse may not be appropriate, the phishing website warning is wrong. However, with the addition to the white list UBPD would be run in the training mode, and this type of false warning will never occur. In fact, the described situation, which could potentially trigger false phishing websites warnings, is very rare, nevertheless it is still good to prevent such false warnings.

Having constructed the initial personal white list, UBPD then presents users with a dialogue which asks for the websites with which users have accounts (shown in Figure 5.3). In the next step UBPD automatically takes users to those websites, and asks users to fill in the authentication forms on those websites one by one. In addition, UBPD also dynamically modifies each authentication webpage, so that all the buttons on the webpage will be replaced with the 'Train(UBPD)' button. Once a user clicks on it, UBPD creates new binding relationship records in the user profile, and then asks whether the user wants to carry on the authentication process (in case it is a multiple step authentication) or go to the next website. A screen shot is shown in Figure 5.4.

It is estimated that on average users have over 20 active web accounts [28]. It would be unrealistic to expect all users to obey instructions and

train UBPD with all their active binding relationships. However, it is reasonable to expect users to train UBPD with their most valuable binding relationships (such as their online accounts with financial organisations). As long as they can do that, their most valuable authentication credentials are protected by UBPD.

5.3.3 Update Of The User Profile:

There are two ways a user profile can be updated with unknown binding relationships. One is initiated by a user and proceeds by the user manually typing in the binding relationships through the provided user interface, and the other is an automatic method which is only active when UBPD is running in the training mode. The former is straightforward, the latter will be described in more detail here.

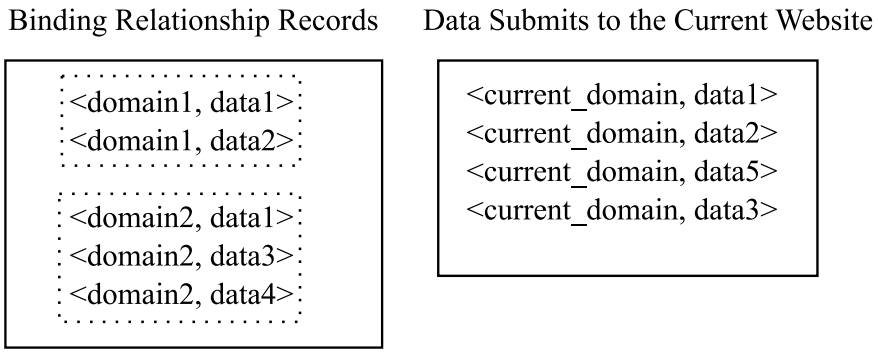
This automatic method tries to detect whether a user is authenticating herself to a server, and if so whether the current binding relationship is known to UBPD or not. If not, the current binding relationships will be added to the user's profile. To detect an authentication session regardless of the nature of the websites involved can be very difficult, as phishing websites could be deliberately implemented to disrupt such detection. Fortunately, with the current design, and especially the help of the personal white list, this authentication session detection method needs only to consider legitimate websites (this method is only active during training mode). It is safe to assume those legitimate websites would not choose to implement their user authentication webpage by using non-standard methods.

The authentication session is detected by analysing the HTML source code, such as the annotation, label, use of certain tags (such as ⟨form⟩) and type of the HTML elements. If the user is using a web authentication form, and the user profile contains no binding relationships with the current website, then UBPD prompts a window to ask the user to update the user profile. If there is an existing binding relationship for the current website, then UBPD will replace the authentication credentials in the binding relationships with the latest values the user submits. If users have entered the authentication credentials wrongly, those credentials will still be stored, but those wrong values will be corrected when users relog in with the correct authentication credentials. In future, the detection of web authentication page usage should be much simpler and more accurate once web authentication interfaces [40, 81, 83] are standardised.

5.3.4 Phishing Score Calculation:

As stated earlier, UBPD would detect only violations of users' binding relationships when UBPD is running in detection mode (e.g. the first user action – visiting a new website, occurs). The violation of binding relationships and the impersonating target of a phishing attack is decided by calculating *phishing scores*.

The calculation is a two step process. In the first step, for each legitimate website with which the user has shared authentication credentials, a temporary phishing score is calculated. Each temporary phishing score is the fraction of the authentication credentials associated with a legitimate website that also appear in the data to be submitted to the current webpage. Its value ranges from 0.0 to 1.0.



P1 : Temporary phishing score for domain1
P2 : Temporary phishing score for domain2
P : The phishing score for the current user-webpage interaction

Step One:
P1 = 2/2 = 1 AND P2 = 2/3= 0.67
Step Two:
P=biggest(P1, P2) = P1= 1.
The target of this phishing attack is users' authentication credentials shared with domain1.

Figure 5.5: An Example of How the Phishing Score Is Calculated

In the second step those temporary phishing scores are sorted into descending order. The current webpage's phishing score is the highest score calculated. The phishing score is a measure of how much authentication credentials have been exposed to potential attackers. Hence, the higher the phishing score, the more likely the user's account will be compromised. The legitimate website with the highest temporary phishing score is considered to be the impersonated target of the phishing website. If more than one legitimate website has yielded the highest phishing score (due to the reuse of authentication credentials), they will all be considered as targets. Although it may not be the attacker's intent, the data they get if an attack succeeds can certainly be used to compromise the user's accounts with those legitimate websites. Figure 5.5 illustrates how a phishing score is calculated in UBPD.

Given the phishing score calculation method, clever attackers may ask victims to submit their credential information through a series of webpages, with each phishing webpage asking only for a small part of data stored in the user profile. To handle this fragmentation attack UBPD has a threshold value and cache mechanism. The system maintains a history of which shared credentials have been released and so when a credential is about to be released an accumulated score can be calculated. Once the phishing score is above the threshold the current webpage will be considered as a phishing webpage. The system's default threshold is 0.6. Why UBPD chooses 0.6 is discussed in section 5.5. If the current phishing score is not zero, UBPD also remembers which data has been sent to the current website and will consider it in the next interaction if there is one. Accordingly many fragmentation attacks can be detected.

5.3.5 Reuse:

It is very common for a user to share the same authentication credentials (user names, passwords, etc) with different websites. So when a user submits the shared authentication credentials to one legitimate website, it could be considered as violation of binding relationship between the authentication credentials and other legitimate websites. The two running modes and the user's personal white list are designed to prevent such false warnings caused by reuse without compromising detection accuracy. A false warning would increase the user's work load, and potentially could lead to USV–D3 vulnerabilities described in Section section 4.5.1.

UBPD detects the violation of binding relationships only when the user is interacting with websites that are neither in the user's white lists nor which the user has account with. So long as legitimate websites for which users have used the same authentication credentials are all contained in the user profile, there will be no false phishing warnings generated due to the reuse. The method that UBPD uses to create the user profile ensures such legitimate websites are most likely to be included, as those websites are either within the user's browsing history or are popular websites in the user's region. Our preliminary false positive evaluation has supported this.

5.3.6 Warning Dialogue:

The warning dialog design is guided by the findings in the user phishing interaction model described in Chapter 3, i.e. to form accurate mental model and make correct decision users need accurate and easy to under-

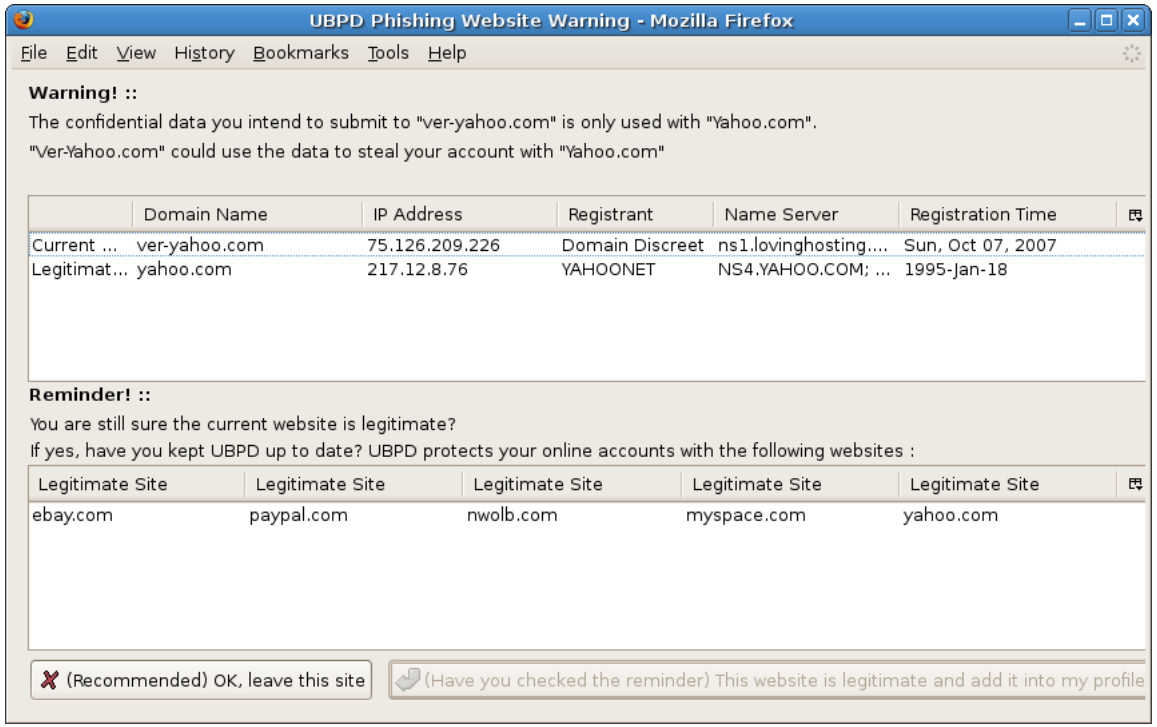


Figure 5.6: Phishing Warning Dialogue

stand messages from the user interface. Figure 5.6 is a warning dialog example. To make the information easy to understand, the dialogue tells users that the current website, to which they are submitting credentials, is not one of the legitimate websites associated with those authentication credentials. To help users understand the detection result and make a correct decision, UBPD identifies up to five areas of differences between the legitimate website and the possible phishing website: the domain name, the domain registrant, the domain registration time, name servers, and IP addresses. Users don't need to understand those terms. They only need be able to recognise the difference between values of the five attributes of the legitimate website and the phishing website.

If the warning dialog is raised due to the reuse of the authentication credentials, then users need to make a decision to train UBPD by clicking the train UBPD button. To make sure users make conscious decisions and this button is disabled by default initially. It can be enabled only by double clicking it. A confirmation dialog, which asks users to specify the account ,whose authentication credentials are reused, is then shown to users. Training will be done only if the correct account is specified.

5.3.7 Website Equivalence:

To discover whether the user is about to submit the authentication credentials to entities with which they have not been bound, UBPD needs to be able to accurately decide whether a given website is equivalent to the website recorded in the user's binding relationships. It is much more complicated than just literally comparing two URLs or IP addresses of two websites, because:

- big organisations often have web sites under different domain names, and users can access their account from any of these domains;

- the IP address of a website can be different each time if dynamic IP addressing is used;

- it is hard to avoid 'pharming' attacks, in which the phishing site's URL is identical to a legitimate one.

UBPD first compares the two websites' domain names and IP addresses. When the two domain names and two IP addresses are equal the web sites are assumed to be identical. Otherwise, the system interrogates the WHOIS[2] database and uses the information returned to determine equivalence. When analysing two different IP addresses our system compares the netnames, name servers, and the countries where each IP address is registered. If they are all identical then the two websites are deemed to be identical. This method can also detect pharming attacks, in which both fraudulent and legitimate websites have the same domain name but are hosted on different IP addresses.

This method is not perfect. A more elegant and complete solution would be a mechanism where servers provide security relevant metadata (including the outsourcing information) to the web browser via a standard protocol as suggested by Behera and Agarwal [8]. However, unless it becomes a standard we have to rely on WHOIS. The Extended Validation Certificate [29] can provide information to decide whether two websites

[2]WHOIS is a TCP-based query/response protocol which is widely used for querying a database in order to determine the owner of a domain name, an IP address. RFC 3912 describes the protocol in detail.

belong to the same party, but the problem is the high cost, high entry level and complexity of obtaining such certificates. Many small and medium businesses will not be able to own them; unless this situation changes they cannot be used to decide website equivalence in general.

5.3.8 User Privacy:

Since a user profile contains confidential user information, it is important that it is secure enough and it does not add new security risks. We use a one-way secure hash function to hash the confidential data before it is stored on the hard disk. When the system needs to determine the equivalence between the data, the system just needs to compare the hash values.

However, because the domain name of the website is not hashed, if the profile is obtained by attackers they would be able to find out with which websites users have accounts and where users have reused their authentication credentials. This information is helpful for attackers; for example, it can be used to launch context aware phishing attacks against the users. To prevent this, when the system is installed it will randomly generate a secret key, and this key will be used to encrypt and decrypt the domain names.

5.3.9 Implementation:

UBPD is implemented as a Firefox add-on for Firefox 2.x. Most of the detection is implemented using JavaScript. The hash function we use is

SHA-1[52] and the encryption method we use is Twofish[1]. The hash function and encryption methods can be changed, we choose to use them mainly because there are open source implementations available. UBPD can easily change to other types of hash and encryption methods when required. The user interface of the system is implemented by using XUL, a technology developed by Mozilla [32]. Although only implemented on Firefox, we are not aware of any impediment to it being ported to other browsers, such as Internet Explorer and Opera. After all, all technology (e.g. JavaScript, algorithms) used in UBPD can be reused on other browsers.

5.4 Evasion And Countermeasures

Once deployed, it is inevitable that attackers would seek vulnerabilities in UBPD that they can use to evade the detection. To prevent such evasions, UBPD is designed with serious consideration of possible evasion techniques. Two types of attack evasion techniques have been considered:

1. Evading detection by manipulating the user submitted data, based on which UBPD checks the violation of the binding relationships. UBPD decides whether a binding relationship is violated by comparing the user submitted data with the authentication credentials that are already shared with legitimate websites. If attackers could somehow make UBPD see different values or a small portion of what a user actually submits, then UBPD will not be able to make correct detection decision.

2. Evading detection by disrupting the detection process. Unlike other phishing websites detection tools, UBPD's detection is not activated until a user submits data to a strange website and its detection warning is raised only shortly before the sensitive data is about to be sent to an attacker. As a result, if attackers manage to delay UBPD's action then their attacks could succeed before users' get warnings.

Below we describe how UBPD is designed to be resilient to these type of evasion techniques.

5.4.1 Manipulating User Submitted Data

The data a user submits to a webpage can be easily retrieved by accessing the DOM interface using client script language, such as JavaScript. Attackers could manipulate the user submitted data before the monitor of UBPD retrieves it.

Disabling the use of any client side script language can solve this problem. However, it does impact users' experience with the Internet. A large number of rich content websites use JavaScripts for client side features. Moreover, many vital business functions provided by today's websites are implemented using such languages; disabling them also disables the functions provided by those websites. As a result, such disabling is not a workable solution. Instead, the monitor of UBPD performs like a key logger. It listens to keyboard and mouse events, and it remembers the data a user has typed into each input field in a webpage. In this way, the data a user submits to a webpage is seen by UBPD first and the webpage

has no chance to manipulate any user submitted data. In general, such key logging is very efficient to run, since it is activated only in detection mode. In training mode, websites that users are visiting are legitimate and the monitor still uses the DOM interface to retrieve the user submitted data, as in those cases client scripts manipulation is not an issue (any website can embed a piece of JavaScript to retrieve user entered data).

5.4.2 Insertion And Fragmentation

UBPD would work perfectly if phishing attacks request the whole set of authentication credentials that a user has shared with the impersonated legitimate website in one phishing webpage. However, attackers may try to vary the amount of sensitive credentials they request in one phishing webpage to disrupt the detection. Below are two possible methods they might use.

Insertion: In this phishing attack, there is only one phishing webpage. The information that the phishing webpage asks for includes not only the authentication credentials, but also includes other data that are not shared between the user and the impersonated website.

Fragmentation: In this phishing attack, there are multiple phishing webpages. On each phishing webpage users will be asked for only a small part of the authentication credentials shared with the impersonated legitimate website.

The current phishing score calculation method (described in section 5.3.4) has been designed to be resilient against insertion attacks. The noise of

the non-related data should not affect the final phishing score.

The problem now is how to avoid fragmentation attacks, and the threshold value is introduced to prevent evasion by fragmentation attacks. Once the phishing score is above the threshold the current webpage will be considered as a phishing webpage. The system's default threshold is 0.6. Whenever a webpage is encountered whose phishing score is not zero, UBPD remembers the sensitive information has been revealed to the website (websites are recognised by their top level domains). In subsequent interactions with this website, the phishing score calculation will also consider the authentication credentials that have been revealed previously. Such phishing score caching will exist only for 24 hours. Accordingly many fragmentation attacks can be detected as well.

5.4.3 Activation Of The Detection Engine

Normally webpages process and forward user submitted data using built-in functions, such as the one provided by the standard web form. (Many legitimate websites use this function to submit the data.) The detection engine is triggered and data submission is suspended when the monitor discovers the use of such functions. This is achieved by listening for 'DOMActivate' [42] events. These events are fired by the browser when such built-in functions have been used. This is the mechanism the monitor uses to activate the detection engine when the system is running in training mode.

However, there are other ways user submitted data could be transmitted to a remote server. Phishers can use client side scripts to implement

these built in functions. For example, by using JavaScript they can access the DOM interface to retrieve the data users enter and use AJAX (Asynchronous JavaScript And XML) techniques to send the data to the server before a user clicks the submit button. If UBPD relied only on listening for 'DOMActivate' events to start the detection decision making, then sensitive user data could have already been sent to attackers. To prevent this evasion, UBPD must monitor the client script function calls when it is running in detection mode. The function calls that should be monitored are 'open()' and 'send()' from the 'xmlhttprequest' API [82]. These are the only two functions that client scripts can use to send data to the server. Once the monitor discovers such function calls, the function is suspended and the detection engine is triggered. Regardless of the data the client scripts would like to send, the detection engine always works on all the data the user has entered to the current webpage. The function call is only resumed if the detection engine thinks the current website is legitimate. In the worst case, attackers would still be able to obtain a part of users' authentication credentials, but they will not be able to gain the full set of user authentication credentials (otherwise the phishing score will be higher than the threshold and an alert will be generated).

5.4.4 Denial Of Service Attack

Since the detection engine would be triggered once the 'xmlhttprequest' API has been called, the attackers can issue this function call continuously to freeze the browser. To prevent this, the monitor can keep a record of whether there is user input since the last time the detection engine was activated for the same webpage. If there is then the detection engine will be activated, otherwise, not.

UBPD decides whether two websites belong to the same entity by analysing the information provided by the WHOIS database. The current implementation uses only the WHOIS database, freely available on the Internet. The attackers could apply Denial of Service attacks on the WHOIS database so that the system would hang. (This is a clear external dependency of our tool.)

5.5 Evaluation

Two experiments have been carried out to evaluate the effectiveness of UBPD in terms of the two following rates:

- **False negative (F_n)** : The probability the system fails to recognise a phishing attack.
- **False positive (F_p)** : The probability the system recognises a legitimate website as a phishing website.

How these two rates are calculated are shown below:

$$F_n = \frac{\text{the number of phishing websites that are not detected}}{\text{total number of websites that are used in the test}}$$

$$F_p = \frac{\text{the number of legitimate websites that are identified as phishing websites}}{\text{total number of websites that are used in the test}}$$

Table 5.1: Characteristics of User Profile

	Alice	Bob	Carol	Dave
Reuse	No	No	Yes	Yes
Uniqueness	Strong	Weak	Strong	Weak

The lower the two rates, the more technically effective is UBPD's detection.

In addition I also search for a useful default threshold value for generating phishing alert (mentioned in section 5.3.4). To focus on evaluating the detection effectiveness, for both experiments UBPD was modified to not present the warning dialogue. Instead it records the phishing score results as well as the URLs for later analysis.

5.5.1 False Negative Rate

From PhishTank [74] and Millersmiles [64] 463 phishing webpages reported between 2nd November 2007 and 16th November 2007 were collected. These phishing websites impersonated Ebay, Paypal, and Natwest bank. I created four user profiles, which describe four artificial users' binding relationships with the three targeted websites. The four user profiles have different characteristics as shown in Table 5.1. 'Reuse' indicates maximum possible reuse of authentication credentials. In this case the user would have same user name and password for Ebay and Paypal. 'Uniqueness' indicates whether the user would use the exact data they shared with a legitimate website at other places. For example if Bob chooses his email address as his password then the uniqueness is weak, because Bob is

Table 5.2: Phishing Websites Characteristics

	Ebay	Paypal	Natwest
AC	211	176	39
AC+PI	22	5	6
PI	4	0	0
Total	237	181	45

very likely to tell other websites his email address. If Bob uses some random string as his password, then the uniqueness is strong, because this random string is unlikely to be used with any other websites. Using different user profiles in this experiment allow us to see whether UBPD's detection will be affected by such user behaviours.

The artificial authentication credentials were submitted to each of the phishing webpages. Regardless of the characteristics of the user profile, the detection result is the same for all four users: 459 pages had a phishing score of 1, and 4 had a phishing score of 0. Thus only four pages evaded detection – F_n is 0.0086.

Detailed analysis confirms that the detection result is determined mainly by the information requested by the phishing webpage. Table 5.2 shows the classification of the phishing webpages based on the type of information they requested. 92% of the collected phishing webpages asked only for authentication credentials and 7.14% of the collected phishing webpages asked both for personal and authentication credentials.

The four phishing web pages UBPD failed to detect asked only for personal information such as full name, address, telephone number and mother's maiden name. In fact, they can not be detected by UBPD no

matter what the threshold value is. However, this is only a minor issue. It is highly suspicious and impractical for phishing attacks to ask for such personal information without asking users to log into their accounts by providing authentication credentials first. In reality phishing websites would normally first present the user with a login webpage before directing the user to the webpage asking for the personal information. In fact, none of the four phishing webpages that UBPD failed to detect are the landing page of the phishing attacks, and all of them are phishing webpages that users would come across after giving up their authentication credentials in previous interaction steps.

5.5.2 False Positive Rate

Five volunteers were provided with the information needed to install UBPD on their machine. They were not explicitly asked to train UBPD with all their binding relationships, because I wanted to see how users would train UBPD and what the false positives would be in reality if the user had not properly trained it. At the end of one week, the result logs were collected from their machines.

The volunteers were three male and two female science students. They all used Firefox as their main web browser. They were all regular Internet users (on average over three hours per day). As a result the UBPD was activated a large number of times and the interactions that occurred during the experiments covered a wide range of types of interaction. Another reason we chose those volunteers is because they are the most unlikely user group to fall victims to phishing attacks [46], because of their technical knowledge and awareness of phishing attacks. Hence,

we can safely assume they have not fallen victims to phishing attacks during the time of study. In total the volunteers interacted with 76 distinct websites, submitted data to those websites 2107 times, and UBPD ran in detection mode only 81 times. In fact all the websites volunteers visited were legitimate. On 59 occasions the phishing score was 0, on five interactions gave a score of 0.25, on 13 occasions the score was 0.5, and the score was 1 on three occasions.

The phishing score was 1 when users interacted with three legitimate websites (the registration webpages of videojug.com and surveys.com, and the authentication webpage of a web forum). The volunteers were then asked what data they supplied to those webpages. It seems that the reuse of authentication credentials on creating new accounts is the reason. In this experiment, the warning dialog is not presented, as we did not aim to test usability. The user must make a decision to train UBPD to remember these new binding relationships, acknowledging that such reuse is not ideal. To avoid the user's confusion about what is the right choice when the warning dialog is presented, the dialog always reminds the user of the legitimate websites UBPD is aware of, and tells the user that if the user is sure the current website is legitimate, and the website is not remembered by UBPD, then they need to update their binding relationships (see the Figure 5.6 in section 5.3.6). This requires no technical knowledge and should be quite easy to understand. There are only two choices provided by the dialog: update the profile and submit the data; or do not send the user submitted data and close the phishing webpage. There is no third choice provided by the dialog, in this way we force the user to make the security decision and they can not just ignore the warnings given by the system.

Many websites force users to supply an email address as the user name.

As a result, the user's email address is kept in the user profile as part of user's authentication credentials. This email address almost inevitably will be given out to other websites, which are not contained in the user profile, for various reasons such as contact method, activate the newly registered account, etc. Thus even when the user does not intend to give out their credentials, the email address nevertheless is shared and UBPD simply confirmed that by calculating the phishing score of 0.5 (which means half of the data the user has shared with a legitimate website was given away to a website that was not in user's profile) on 13 occasions.

For one volunteer five interactions gave a phishing score of 0.25. The user had an account at a major bank, the authentication credentials for which compromised four data items. One of these was the family name. For other sites not included in the user's profile asking for this information caused our system to identify the sharing of the data.

Based on the figures from both experiments I decided to set the default threshold value to 0.6. First, it can successfully detect phishing webpages asking for more than half of the credentials the user has shared with a legitimate website (99.14% of the phishing websites in Experiment One can be detected). It also generated few false positives. The false positive rate of the system is 0.0014 (obtained by dividing the number of false positives generated with the total number of times the UBPD was activated).

The result of the false positive evaluation shows UBPD has a small false positive rate, and it also shows that the reuse of the authentication credentials and partial training are the main cause of the false positives.

5.6 Discussion

5.6.1 Why UBPD Is Useful

Besides its high detection accuracy, UBPD is useful also because it complements existing detection systems. First UBPD detects phishing websites based on users' behaviours, not the incoming data that attackers can manipulate freely. Violation of the binding relationships cannot be changed no matter what techniques phishers choose to use. As the evaluation proves, UBPD is able to consistently detect phishing webpages regardless of how they are implemented as long as they ask for authentication credentials. In contrast detection systems based on incoming data may find it difficult to deal with novel and sophisticated spoofing techniques. UBPD analyses the identity of websites using both IP addresses and domain names, it can detect pharming attacks, which are undetectable by many existing systems. Being independent of the incoming data means low cost in maintenance, the system does not need updating when attackers vary their techniques, and so we have far fewer evasion techniques to deal with.

Some systems have tried to stop phishing attacks from reaching the users (phishing site take down, botnet take down, phishing email filter, etc.), some have tried to detect phishing webpages as soon as users arrive at a new web page (Phishing URL blacklist, netcraft toolbar, spoofguard, CARTINA, etc), and some have tried to provide useful information or indicators to help users to detect phishing websites. However, there are no other systems that work at the stage when phishing webpages have somehow penetrated through and users have started to give out information

to them. Thus UBPD can complement existing techniques: if phishing attacks evade earlier detection, then UBPD provides an additional tool for intervention. There seems little reason to believe UBPD cannot be effectively combined with other tools.

Finally UBPD focuses on the release of credentials. It is agnostic as to how the user reached the point of attempted release.

5.6.2 Performance

The two working modes of UBPD reduce the computing complexity. Detection mode consumes more computing power than training mode, but it runs only when users are interacting with the websites that are not contained in the user profile (in our second experiment UBPD only in detection mode 81 out of 2107 times). Computing in detection mode is still light weight for the computing power of an average personal computer. (None of the volunteers noticed any delay.) As a result, UBPD is efficient to run and adds little delay to the existing user-webpage interaction experience.

5.6.3 Limitations And Future Work

UBPD should be viewed as an initial but promising effort towards detecting phishing by analysing user behaviour. Despite its detection effectiveness, there are some limitations within the implementation and design. These are discussed below.

Currently UBPD cannot handle all types of authentication credentials. It can handle static type authentication credentials such as user name, password, security questions, etc, but dynamic authentication credentials shared between users and the legitimate websites cannot be handled by the current implementation (e.g. one-time passwords). In future, it would be useful to be able to handle such credentials.

There is also another method for a phisher to evade detection. Attackers might compromise a website within a user's personal white list and host the phishing website under the website's domain (perhaps for several hours to a couple of days). This method is difficult to carry out, and it is unlikely to happen if there are still easier ways to launch phishing attacks.

Detection itself requires little computing time, in fact, retrieving the data from WHOIS database is the most time consuming part. It depends on the type of network the user is using and the workload on the WHOIS database. In future cache functionality could be added to reduce the number of queries for each detection. Currently the system produces a slight delay because of the WHOIS database query.

Each user profile contains useful security information, such as the websites the user has binding relationships with. Once deployed, UBPD is likely to discover active phishing websites at the earliest stages. In future it would be very interesting to investigate how this information can be collected safely and used to provide more accurate and timely detection.

A more thorough and large scale study to compare the detection effectiveness of UBPD and other major existing phishing detection tools would

provide further evidence that the detection by UBPD is more accurate and can not be affected by the variation of attack techniques.

Improvements to the method to determine the value of the phishing score threshold could be made. Essentially the idea of having a threshold value is to alert users when a large portion of their authentication credentials are about to be shared with third parties. Different user accounts require different numbers of authentication credentials, so the current approach to have a universal threshold value may be too sensitive to some accounts but in the mean time may not be secure enough for others.

One potential approach is to make the threshold value specific to individual user accounts, i.e. the threshold value is dependent on the number of authentication credentials for the phishing attack target. For example for an account with two authentication credentials this value could be 1.0; and for an account with three authentication credentials this value can be 0.66. The threshold value can be chosen at the run time based on the number of authentication credentials the predicted target account has (UBPD can predict the target of phishing attack based on the credentials users are about to release).

5.7 Conclusion

Surveys of literature (in chapter 2) revealed that all existing anti-phishing techniques concentrate on detecting the attack, not on the actual release of credentials. It is the release of credentials, that is the actual problem. UBPD focuses on this feature of phishing attacks, and does not rely on

detection of any specific means by which the user has been persuaded to release such data(which may take many formats). The detection approach has unique strength in protecting users from phishing threats, and it also makes the detection system hard to circumvent. UBPD is not designed to replace existing techniques. Rather it should be used to complement other techniques, to provide better overall protection. The detection approach used in UBPD fills a significant gap in current anti-phishing technology capability.

Chapter 6

Evaluations and Future Work

6.1 The Hypothesis

This research reported in the previous chapters provides evidence in support of the following proposition:

> **A more refined understanding of the nature of deception in phishing attacks would facilitate more effective user-centred threat identification of web based authentication systems, the development of countermeasures to identified threats, and the production of guidelines for phishing–resilient system designs.**

Below the achievements of each technical chapter in this thesis are identified and assessed. The text below also indicates novel aspects of the work performed.

6.2 Evaluation

6.2.1 Understanding Of The Nature Of Deception In Phishing Attack And The Model Of User Phishing Interaction

The research described in Chapter 3 exhibited considerable originality and achievement in increasing understanding of the nature of deception in phishing attacks. The following contributions have been made:

Creation of a social engineering attack incident data set. This data set is the only social engineering attack incident collection on the Internet. It covers a comprehensive range of attack techniques, attack delivery channels, technical vulnerabilities exploited, human factors involved, etc. It can be used for analysing the nature of social engineering attacks and for other purposes, such as security awareness user education.

Improved understanding and interpretation of phishing attacks from a decision making point of view. The user's decision making process during a phishing attack encounter has been analysed. The user's ability to generate better solutions and evaluate available options is found to be secondary in deciding the outcome of an attack. The most important factor is how accurate is the perception of the current situation.

Furthermore, we can see that an attacker tries to engineer a false perception within the victim's mind, and also tries to simplify the solution generation stage by telling users what actions he/she should take to respond to the false perception. He or she must now decide only whether to take the suggested means of solving the problem. The suggested actions

are rational given the false perception.

Creation of a framework for predicting users' behaviours during phishing encounters. A graphical model to capture the process of user-phishing interaction and important factors within this process is abstracted. This model focuses on how users construct perceptions and describes how each important factor could influence an attack outcome.

A systematic answer the question: why do users fall victims to phishing attacks? With reference to the graphical model, this question is answered comprehensively and in a systematic manner. In contrast, other research has answered this question in an ad hoc manner (carrying out user experiments in controlled environments). The answers produced by my method are more comprehensive.

6.2.2 Guidelines For Phishing Attack Countermeasures

In Chapter 3, based on the improved and refined understanding of phishing attacks, guidelines for designing more usable and effective security indicators/tools, developing effective security education and evaluating security indicators/tools were suggested.

6.2.3 Threat Modelling For Web Based User Authentication Systems

In Chapter 4, a framework and related methods to identify, and assess user-related vulnerabilities within internet based user authentication

systems are introduced. This threat modelling method takes an assets-centric approach and has the following four steps:

1. **Asset identification**: identify all the valuable assets (authentication credentials which can be used to prove a user's identity) that attackers might obtain from users.

2. **Threat Identification**: identify threats the authentication credentials (assets) face based on the properties of these authentication credentials.

3. **Vulnerability and risk level assessment**: for each threat identified apply the corresponding vulnerability analysis technique to reveal the potential vulnerabilities and associated risk levels.

4. **Threat mitigation**: determine the countermeasures for each vulnerability identified.

This technique is designed to be quantifiable, and can be easily integrated with existing standard vulnerability analyses and risk assessments. Two cases studies have been used to illustrate how the method should be applied.

6.2.4 Phishing Websites Detection Tools

The user behaviour based phishing websites detection solution described in Chapter 5 shows significant novelty. As a previous part of the research revealed, the construction of a false perception is key to the success of a

phishing attack. There are two possible approaches to detect attackers' efforts to engineer a false perception:

- analyse the content of the websites and other relevant incoming data (from attackers to user victims).
- monitor the occurrence of user behaviours which are the result of successful phishing attacks.

The latter approach has many advantages. It needs to deal with much less low level technical detail, has lower maintenance costs, and is fundamentally resilient against a variety of attack evasion techniques. It has been used to develop the phishing websites detection prototype. The evaluation of the detection effectiveness has shown that both high detection and very low false positive rates are achieved.

6.3 Future Work

6.3.1 Refine And Improve The User-Phishing Interaction Model

The user-phishing interaction model was derived from application of cognitive walkthroughs. A large scale controlled user study and follow on interviews could be carried out to provide a more rigorous conclusion.

The current model does not describe irrational decision making nor address influence by other external factors such as emotion, pressure, and

other human factors. It would be very useful to expand the model to accommodate these factors.

6.3.2 Study Special User Groups

Current research in human factors in security has not targeted any specific user group. Many research studies even claim that age, agenda, and education background have little influence on the results of phishing attacks. I think it would be interesting to challenge those notions. Children, elderly, and disabled user groups are very likely to have distinct features from the general population, e.g., differences in their perception of security and privacy, knowledge of information systems, ability to construct accurate perceptions and give adequate attention to security. Given their vulnerable nature, research concerning these user groups is needed. They are obvious 'soft targets' for the professional phishers. They may need specific solutions to protect them from falling victim to phishing attacks.

6.3.3 Insider Phishing Attacks

In general, attacks from insiders are a common and very serious threat to information systems. Until now, most reported phishing attacks are from outsiders. Insiders, who have personal knowledge of and relationships with victims, can also use phishing attacks to achieve their goals. They can easily launch speared phishing attacks that are most effective. Research is needed to determine how one can protect users' from such threats without compromising trust among users.

6.3.4 Usable Authentication Methods

Despite many efforts, there is still a lack of usable, low cost, secure user authentication methods. Numerous multi-factor and multi-channel authentication methods have been proposed. They often add little security to the user authentication system, and many of them require substantial financial investment to implement and deploy, or have operational constraints. Also, the usability of these new authentication methods are yet to be evaluated.

6.3.5 Improving The User Behaviours Based Phishing Website Detection

The current phishing website detection prototype can detect only phishing websites that request static authentication credentials such as a password. It would be useful to research how dynamic authentication credentials, such as one time passwords, can be securely dealt with.

Another improvement that could be made is to take advantage of a large collection of individual user profiles. For example, the phishing websites detected by individual users could be quickly reported, the phishing websites could be taken down quickly, and a more timely black list could be built.

Applying this detection approach to detect other social engineering attacks on the Internet is also a possibility.

6.3.6 Conclusion

Finally, phishing attacks are a major problem. It is important that they are countered. The work reported in this thesis indicates how understanding of the nature of phishing may be increased and provides a method to identify phishing problems in systems. It also contains a prototype of a system that catches those phishing attacks that evaded other defences, i.e. those attacks that have "slipped through the net". An original contribution has been made in this important field, and the work reported here has the potential to make the internet world a safer place for a significant number of people.

Appendix A

Appendix

A.1 Cognitive Walkthrough Example

As described in chapter 3, the cognitive walkthrough method has been used to analysing how users make decisions when interacting with phishing attacks. An example of the cognitive walkthrough is provided to help readers understand how it is performed.

The process of walkthrough has four steps (see details at subsection 3.1.3). The first three steps are described below, the fourth step is simply to record the findings and is not described.

A.1.1 Input

There are four aspects of the inputs to be defined. As described in subsection 3.1.3, users and interfaces are common for each walkthourgh.

The users are the general public and interfaces are the most common software tools. (In this case, it is IE7 and Firefox for web browsing, and Outlook Express for email clients). For this particular walkthrough, sample tasks (an attacking incident) and action sequences are described below:

Phishing Attack incident

The targets of this phishing attack are eBay users. The attack engages users via email in which the embedded URL link could lead users to a phishing website. The phishing website then harvests users' authentication credentials. Sellers on eBay website receive questions from potential buyers frequently, the attacker in this incident prepared a phishing email which looks like an automated email sent by Ebay as a result of a buyer's question. Below is the screen shot of the phishing email. The attack was archived by Millersmiles[2], and the incident was reported on 29th January 2009.

Action Sequence

Since the purpose of the walkthrough is to understand the decision making leading to satisfaction of the attacker's goal, the following sequences are identified:

- First, the victim reads the header of the email, i.e. the subject title and sender's email address.

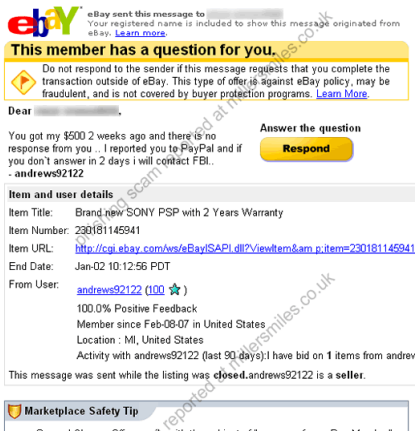


Figure A.1: Phishing Email Screen Shot

- Second, the victim reads the content of the email and clicks the link embedded with the intention of answering the buyer's question.

- Third, having arrived at the phishing website, the victim finds that he/she needs to log into their account first. Then he/she types in their user name and password.

A.1.2 Walkthrough And Analysis

The analysis starts with the first action. For each action the questions listed in section subsection 3.1.3 are asked.

Questions and Answers for Action One:

- What is the context of the user phishing attack interaction? Answer:

The victim is an eBay user and is currently selling products on eBay.

- What are the user's assumptions or expectations? Answer: The victim anticipates that potential buyers could ask him/her questions and is expecting to receive question emails from eBay.

- What is the false perception attackers try to engineer? Answer: A genuine buyer has a question to ask.

- What would be victims' expected feedback from the user interface for each action they have taken? Answer: The victim expects the email client to display the email title and sender's email address. The sender's email address should indicate it is from eBay.

- What information **obviously** available at the user interface can be selected/interpreted to form a false perception? Answer: Both title and sender's email address.

- What information **obviously** available at the user interface can be selected/interpreted to form an accurate perception? Answer: None

- Would users know what to do if they wished to determine the authenticity of the party they are interacting with. If so, is it easy to perform the check consistently? Answers: Since the user lacks technical knowledge of the implementation of the email system, they would not know how to check the authenticity of the sender at this stage.

Questions and Answers for Action Two:

- What is the context of the user phishing attack interaction? Answer: The victim is an Ebay user and is currently selling products on eBay.

- What are the user's assumptions or expectation? Answer: The victim expects the content of the email to contain the question from the buyer regarding the product that the victim is currently selling.

- What is the false perception attackers try to engineer? Answer: A genuine buyer has a question to ask.

- What would be victims' expected feedback from the user interface for each action they have taken? Answer: The victim expects the email client to display the content of the email; the style and format of the email should be consistent with the previous email he/she received from Ebay; the email should address the victim using his/her user name; there should be a button to respond to the question immediately.

- What information **obviously** available at the user interface can be selected/interpreted to form a false perception? Answer: The entire content of the email.

- What information **obviously** available at the user interface can be selected/interpreted to form an accurate perception? Answer: None

- Would users know what to do if they wished to determine the authenticity of the party they are interacting with. If so, is it easy to perform the check consistently? Answers: The URL link of the respond button is http://signin.eBay.com.411-563-829.411-563-829.j5rljp7hvtzh.hx5rb9vcuhn.info/sc/saw-cgi/eBayISAPI.dll/, which does not

belong to eBay. However, the user, who lacks technical knowledge regarding the implementation of the URL naming system, would not be able to consistently interpret the deceiving URL correctly. The action to check the URL for the button is also not convenient, so it may not be performed consistently even if users know how to.

Questions and Answers for Action Three:

- What is the context of the user phishing attack interaction? Answer: The victim is an Ebay user and is currently selling products on eBay.
- What are the user's assumptions or expectations? Answer: The victim expects to visit eBay website and log into their account.
- What is the false perception attackers try to engineer? Answer: A genuine buyer has a question to ask and the victim now is visiting the Ebay website.
- What would be victims' expected feedback from the user interface for each action they have taken? Answer: The victim expects the web browser to display the official eBay website.
- What information **obviously** available at the user interface can be selected/interpreted to form a false perception? Answer: The content of the webpage displayed, including its template and style.
- What information **obviously** available at the user interface can be selected/interpreted to form an accurate perception? Answer: The URL address displayed in the URL bar. Note: the lack of certificate

should not be considered as obviously available, as users are not good at noticing something missing from the interfaces.

- Would users know what to do if they wished to check the authenticity of the counterpart they are interacting with? If so, is it easy to perform the check consistently? Answers: The same as for Action Two. The victim lacks the knowledge to do so. To check the URL address and digital certificate is easy to perform, however, since there is no enforcement, users may not be able to do this consistently all the time.

Bibliography

[1] Javascript encryption program. http://home.versatel.nl/MAvanEverdingen/Code/. Last accested 6th March 2011.

[2] Millersmiles home page. http://www.millersmiles.co.uk/. Last accessed on 6th March 2011.

[3] Spamassassin's official website. http://spamassassin.apache.org/. Last accessed on 6th March 2011.

[4] Department of defense standard : Trusted computer system evaluation criteria. DoD 5200.28-STD Supersedes CSC-STD-001-83, dtd l5 Aug 83, December 1985.

[5] information technology - security techniques - information security management systems - requirements. *International Organization for Standardization and the International Electrotechnical Commission*, 2007.

[6] C. Abad. The economy of phishing: A survey of the operations of the phishing market. *First Monday*, 10(9), 2005.

[7] K. Albrecht, N. Burri, and R. Wattenhofer. Spamato - An Extend-

able Spam Filter System. In *2nd Conference on Email and Anti-Spam (CEAS), Stanford University, Palo Alto, California, USA*, July 2005.

[8] P. Behera and N. Agarwal. A confidence model for web browsing. In *Toward a More Secure Web - W3C Workshop on Transparency and Usability of Web Authentication*, 2006.

[9] T. Betsch and S. Haberstroh, editors. *The routines of decision making*. Mahwah, NJ ; London : Lawrence Erlbaum Associates., 2005.

[10] A. Bortz and D. Boneh. Exposing private information by timing web applications. *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 621–628, 2007.

[11] J. Brainard, A. Juels, R. L. Rivest, M. Szydlo, and M. Yung. Fourth-factor authentication: somebody you know. In *CCS '06: Proceedings of the 13th ACM conference on Computer and communications security*, pages 168–178, New York, NY, USA, 2006. ACM.

[12] K. Cameron and M. B. Jones. Design rationale behind the identity metasystem architecture. Technical report, Microsoft, 2006.

[13] W. Cathleen, J. Rieman, L. Clayton, and P. Peter. The cognitive walk-through method: A practitioner's guide. Technical report, Institute of Cognitive Science, University of Colorado, 1993.

[14] M. Chandrasekaran, R. Chinchain, and S. Upadhyaya. Mimicking user response to prevent phishing attacks. In *IEEE International Symposium on a World of Wireless, Mobile, and Multimedia networks*, 2006.

[15] charles cresson wood. *Information Security Policies Made Easy Version 8*. InfoSecurity Infrastructure, 2001.

[16] N. Chou, R. Ledesma, Y. Teraguchi, D. Boneh, and J. C. Mitchell. Client-side defense against web-based identity theft. In *NDSS '04: Proceedings of the 11th Annual Network and Distributed System Security Symposium*, February 2004.

[17] F. T. Commission. An e-card for you game. http://www.ftc.gov/bcp/conline/ecards/phishing/index.html. Last Accessed in 2009.

[18] F. T. Commission. Phishing alerts. http://www.ftc.gov/bcp/conline/pubs/alerts/phishingalrt.htm. Last accessed in 2009.

[19] S. S. Consulting. Detecting states of authentication with protected images. http://ha.ckers.org/blog/20061108/detecting-states-of-authentication-with-protected-images/, Novermber 2006. Last accessed on 6th March 2011.

[20] DarkReading. Social engineering, the usb way. http://www.darkreading.com/security/perimeter/showArticle.jhtml?articleID=208803634, June 2006. Last accessed on 6th March 2011.

[21] R. Dhamija, D. Tygar, and M. Hearst. Why phishing works. *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*,, ACM Special Interest Group on Computer-Human Interaction:581–590, 2006.

[22] R. Dhamija and J. D. Tygar. The battle against phishing: Dynamic security skins. In *SOUPS '05: Proceedings of the 2005 symposium on*

Usable privacy and security, pages 77–88, New York, NY, USA, 2005. ACM.

[23] X. Dong, J. A. Clark, and J. Jacob. Modelling user-phishing interaction. In *Conference on Human System Interaction*, 2008.

[24] J. S. Downs, M. B. Holbrook, and L. F. Cranor. Decision strategies and susceptibility to phishing. In *SOUPS '06: Proceedings of the second symposium on Usable privacy and security*, pages 79–90, New York, NY, USA, 2006. ACM Press.

[25] Ebay. Spoof email tutorial. http://pages.ebay.com/education/spooftutorial/. Last accessed on 6th March 2011.

[26] S. Egelman, L. F. Cranor, and J. Hong. You've been warned: an empirical study of the effectiveness of web browser phishing warnings. *CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, pages 1065–1074, 2008.

[27] I. Fette, N. Sadeh, and A. Tomasic. Learning to detect phishing emails. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 649–656, New York, NY, USA, 2007. ACM Press.

[28] D. Florencio and C. Herley. A large-scale study of web password habits. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 657–666, New York, NY, USA, 2007. ACM Press.

[29] R. Franco. Better website identification and extended validation certificates in ie7 and other browsers. IEBlog, November 2005.

[30] M. Frontier. Phishing iq. http://survey.mailfrontier.com/survey/quiztest.html. Last Accessed on 6th March 2011.

[31] A. Giani and P. Thompson. Detecting deception in the context of web 2.0. In *Web 2.0 Security & Privacy 2007*, 2007.

[32] B. Goodger, I. Hickson, D. Hyatt, and C. Waterson. Xml user interface language (xul) 1.0. Technical report, Mozilla Org., 2001.

[33] D. Gragg. A multi-level defense against social engineering. *SANS Institute*, GSEC Option 1 Version 1.4b, 2003.

[34] S. Granger. *Social Engineering Fundamentals, Part I: Hacker Tactics*. Security Focus, Dec. 2001.

[35] S. Granger. Social engineering fundamentals, part ii: Combat strategies security focus, Jan 2002.

[36] S. Granger. Social engineering reloaded. http://www.symantec.com/connect/articles/social-engineering-reloaded, March 2006. Last Accessed on 6th March 2011.

[37] S. Grazioli. Where did they go wrong? an analysis of the failure of knowledgeable internet consumers to detect deception over the internet. *Group Decision and Negotiation*, 2:149 172, 2004.

[38] V. Griffith and M. Jakobsson. Messin' with texas, deriving mother's

maiden names using public records. *Applied Cryptography and Network Security: Third International Conference, Lecture Notes in Computer Science*, 3531, June 2005.

[39] Harl. People hacking: The psychology of social engineering. Text of Harl's Talk at Access All Areas III, July 1997.

[40] S. Hartman. Ietf-draft: Requirements for web authentication resistant to phishing. Technical report, MIT, 2007.

[41] A. H. Hastorf and H. Cantril. They saw a game: A case study. *Journal of Abnormal and Social Psychology*, 49:129–134, 1954.

[42] B. Höhrmann, P. L. Hégaret, and T. Pixley. Document object model events. Technical report, W3C, 2007.

[43] IEBlog. Better website identification and extended validation certificates in ie7 and other browsers. http://blogs.msdn.com/b/ie/archive/2005/11/21/495507.aspx, 2005. Last accessed on 6th March 2011.

[44] E. Inc. Ebay toolbars. http://pages.ebay.co.uk/ebay_toolbar/tours/index.html. Last Accessed 6th March 2011.

[45] C. Jackson, D. R. Simon, D. S. Tan, and A. Barth. An evaluation of extended validation and picture in picture phishing attacks. In *In Proceedings of Usable Security (USEC'07)*, 2007.

[46] T. Jagatic, N. Johnson, M. Jakobsson, and F. Menczer. Social phishing. *ACM Communication*, 50:94–100, October 2007.

[47] M. Jakobsson. Modeling and preventing phishing attacks. In *Phishing Panel in Financial Cryptography '05*, 2005.

[48] M. Jakobsson. The human factors in phishing. *Privacy & Security of Consumer Information '07*, 2007.

[49] M. Jakobsson and J. Ratkiewicz. Designing ethical phishing experiments: a study of (rot13) ronl query features. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 513–522, New York, NY, USA, 2006. ACM Press.

[50] M. Jakobsson, A. Tsow, A. Shah, E. Blevis, and Y.-K. Lim. What instills trust? a qualitative study of phishing. In *Extended abstract, USEC '07*, 2007.

[51] I. Janis and L. Mann. *Decision making: a psychological analysis of conflict, choice, and commitment*. Free Press, 1979.

[52] P. A. Johnston. http://pajhome.org.uk/crypt/index.html. Last accessed 6th March 2011.

[53] A. Josang, B. AlFayyadh, T. Grandison, M. AlZomai, and J. McNamara. Security usability principles for vulnerability analysis and risk assessment. *Computer Security Applications Conference, Annual*, 0:269–278, 2007.

[54] A. Klein. In session phishing attacks. *Trusteer Research Paper*, 2008.

[55] G. Klein. *Source of Power: How people make decisions*. MIT Press, Cambridge, MA, 1998.

[56] P. C. Kocher. Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems. pages 104–113. Springer-Verlag, 1996.

[57] P. Kumaraguru, Y. Rhee, A. Acquisti, L. F. Cranor, J. Hong, and E. Nunge. Protecting people from phishing: the design and evaluation of an embedded training email system. pages 905–914, 2007.

[58] A. Litan. Toolkit: E-commerce loses big because of security concerns. Technical report, Garnter Research, 2006.

[59] J. Long and J. Wiles. *No Tech Hacking: A Guide to Social Engineering, Dumpster Diving, and Shoulder Surfing*. Syngress, 2008.

[60] T. McCall. Gartner survey shows phishing attacks escalated in 2007; more than $3 billion lost to these attacks. Technical report, Gartner Research, 2007.

[61] D. McMillen, S. Smith, and E. Wells-Parker. The effects of alcohol, expectancy, and sensation seeking on driving risk taking. *Addictive Behaviours*, 14:477–483, 1989.

[62] Microsoft. Consumer awareness page on phishing. http://www.microsoft.com/athome/security/email/phishing.mspx. Last accessed 6th March 2011.

[63] Microsoft. Anti-phishing white paper. Technical report, Microsoft, 2005.

[64] MillerSmiles. Official website. http://www.millersmiles.co.uk. Last Accessed 6th March 2011.

[65] K. D. Mitnick, W. L. Simon, and S. Wozniak. *The Art of Deception: Controlling the Human Element of Security*. Wiley, 2002.

[66] T. Moore and R. Clayton. Examining the impact of website take-down on phishing. In *eCrime '07: Proceedings of the anti-phishing working groups 2nd annual eCrime researchers summit*, pages 1–13, New York, NY, USA, 2007. ACM.

[67] Mozilla. Phishing protection. http://www.mozilla.com/en-US/firefox/phishing-protection/, 2007. Last Accessed on 6th March 2011.

[68] Netcraft. http://toolbar.netcraft.com/, 2007. Last accessed on 6th March 2011.

[69] G. Ollmann. The pharming guide. Technical report, Next Generation Security Software Ltd., 2005.

[70] G. Ollmann. The phishing guide. Technical report, NGSS, 2005.

[71] M. Org. Firefox 2 phishing protection effectiveness testing. http://www.mozilla.org/security/phishing-test.html, Nov. 2006.

[72] Y. Pan and X. Ding. Anomaly based web phishing page detection. *acsac*, 0:381–392, 2006.

[73] A. phishing work group. Anti-phishing work group home page. http://www.antiphishing.org/. Last accessed on 6th March 2011.

[74] Phishtank. http://www.phishtank.com/, 2007. Last accessed on 6th March 2011.

[75] S. Plous. *The Psychology of Judgment and Decision Making*. Number ISBN-10: 0070504776. McGraw-Hill, 1993.

[76] J. J. RUSCH. The "social engineering" of internet fraud. In *Internet Global Summit*, 1999.

[77] S. Schechter, R. Dhamija, A. Ozment, and I. Fischer. The emperor's new security indicators: An evaluation of website authentication and the effect of role playing on usability studies. In *2007 IEEE Symposium on Security and Privacy*, 2007.

[78] F. Schick. *Making choices : a recasting of decision theory*. New York : Cambridge University Press, 1997.

[79] R. Security. Enhancing one-time passwords for protection against real-time phishing attacks. Technical report, RSA, 2007.

[80] H. A. Simon. *Models of man: social and rational: mathematical essays on rational human behavior in a social setting*. Wiley, 1957.

[81] G. Staikos. Web browser developers work together on security. Web, November 2005.

[82] A. van Kesteren. The xmlhttprequest object. Technical report, W3C, 2006.

[83] W3C. Web security context - working group charter. http://www.w3.org/2005/Security/wsc-charter, 2006. Last accessed on 6th March 2011.

[84] D. Watson, T. Holz, and S. Mueller. Know your enemy: Phishing. Technical report, The Honeynet Project & Research Alliance, 2005.

[85] G. Wilson and D. Abrams. Effects of alcohol on social anxiety and physiological arousal: Cognitive versus pharmacological processes. *Cognitive Research and Therapy*, 1:195–210, 1977.

[86] M. S. Wogalter, editor. *Handbook of Warnings*. Lawrence Erlbaum Associates, 2006.

[87] R. Wright, S. Chakraborty, A. Basoglu, and K. Marett. Where did they go right? understanding the deception in phishing communications. *Group Decision and Negotiation*, 19:391 416, 2010.

[88] M. Wu, R. C. Miller, and S. L. Garfinkel. Do security toolbars actually prevent phishing attacks? In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 601–610, New York, NY, USA, 2006. ACM Press.

[89] M. Wu, R. C. Miller, and G. Little. Web wallet: preventing phishing attacks by revealing user intentions. *SOUPS '06: Proceedings of the second symposium on Usable privacy and security*, pages 102–113, 2006.

[90] Y. Zhang, S. Egelman, L. Cranor, and J. Hong. Phinding phish: Evaluating anti-phishing tools. In *In Proceedings of the 14th Annual Network and Distributed System Security Symposium (NDSS 2007)*, page 2007, 2007.

[91] Y. Zhang, J. I. Hong, and L. F. Cranor. Cantina: a content-based approach to detecting phishing web sites. In *WWW '07: Proceedings*

of the 16th international conference on World Wide Web, pages 639–648, New York, NY, USA, 2007. ACM Press.